Optimization for Online Platforms

by

Deeksha Sinha

B.Tech and M.Tech., Indian Institute of Technology Bombay (2014)

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2021

Author .....
Sloan School of Management
January 7, 2021

Certified by .....
Vivek. F. Farias
Patrick J. McGovern (1959) Professor
Thesis Supervisor

Accepted by .....
Georgia Perakis
William F. Pounds Professor of Management
Co-director, Operations Research Center

Optimization for Online Platforms

by

Deeksha Sinha

Submitted to the Sloan School of Management
on January 7, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

In the last decade, there has been a surge in online platforms for providing a wide variety of services. These platforms face an array of challenges that can be mitigated with appropriate modeling and the use of optimization tools. In this thesis, we examine, model, and provide solutions to some of the key challenges.

First, we focus on the problem of intelligent SMS routing faced by several online platforms today. In a dynamically changing environment, platforms need to carefully choose SMS aggregators to have a high number of text messages being delivered to users at a low cost. To model this problem, we consider a novel variant of the multi-armed bandit (MAB) problem, *MAB with cost subsidy*, which models many real-life applications where the learning agent has to pay to select an arm and is concerned about optimizing cumulative costs and rewards. We show that naive generalizations of existing MAB algorithms like Upper Confidence Bound and Thompson Sampling do not perform well for the SMS routing problem. For an instance with K arms and time horizon T, we then establish a fundamental lower bound of $\Omega(K^{1/3}T^{2/3})$ on the performance of any online learning algorithm for this problem, highlighting the hardness of our problem in comparison to the classical MAB problem. We also present a simple variant of *explore-then-commit* and establish near-optimal regret bounds for this algorithm. Lastly, we perform numerical simulations to understand the behavior of a suite of algorithms for various instances and recommend a practical guide to employ different algorithms.

Second, we focus on the problem of making real-time personalized recommendations which are now needed in just about every online setting, ranging from media platforms to e-commerce to social networks. While the challenge of estimating user preferences has garnered significant attention, the operational problem of using such preferences to construct personalized offer sets to users is still a challenge, particularly in modern settings with a massive number of items and a millisecond response time requirement. Thus motivated, we propose an algorithm for personalized offer set optimization that runs in time sub-linear in the number of items while enjoying a uniform performance guarantee. Our algorithm works for an extremely general class of problems and models of user choice that includes the mixed multinomial logit model as a special case. Our algorithm can be entirely data-driven and empirical evaluation on a massive content discovery dataset shows that our implementation indeed runs fast and with increased performance relative to existing fast heuristics.

Third, we study the problem of modeling purchase of multiple items (in online and offline settings) and utilizing it to display optimized recommendations, which can lead to significantly higher revenues as compared to capturing purchase of only a single product in each transaction. We present a parsimonious multi-purchase family of choice models called the BundleMVL-K family, and develop a binary search based iterative strategy that efficiently computes optimized recommendations for this model. We establish the hardness of computing optimal recommen-

dation sets and characterize structural properties of the optimal solution. The efficacy of our modeling and optimization techniques compared to competing solutions is shown using several real-world datasets on multiple metrics such as model fit, expected revenue gains, and run-time reductions.

Fourth, we study the problem of A-B testing for online platforms. Unlike traditional offline A-B testing, online platforms face some unique challenges such as sequential allocation of users into treatment groups, large number of user covariates to balance, and limited number of users available for each experiment, making randomization inefficient. We consider the problem of optimally allocating test subjects to either treatment with a view to maximize the precision of our estimate of the treatment effect. Our main contribution is a tractable algorithm for this problem in the online setting, where subjects must be assigned as they arrive, with covariates drawn from an elliptical distribution with finite second moment. We further characterize the gain in precision afforded by optimized allocations relative to randomized allocations and show that this gain grows large as the number of covariates grows.

Thesis Supervisor: Vivek F. Farias
Title: Patrick J. McGovern (1959) Professor

Acknowledgements

I would like to begin by thanking Vivek Farias, my thesis advisor, for his mentorship and support in my journey as a PhD candidate. His enthusiasm for finding and solving meaningful research problems is truly infectious. His openness to exploring questions in new domains has significantly broadened my horizons. I am very grateful for his flexibility in letting me pursue different research directions. Most importantly, his emphasis on the potential impact of a problem will always serve as a rudder for me.

I was also lucky to be able to work on research projects with all my thesis committee members - Jónas Oddur Jónasson, Andrew Li and Nikos Trichakis. Andrew's focus on intuition and Nikos' and Jonas' systematic approach to seemingly uncertain research problems are some qualities I wish to emulate in my career as a researcher. I also wish to thank Restef Levi and Tauhid Zaman. Each, with their unique teaching style, made my experience as a teaching assistant valuable.

In addition to my advisor and thesis committee members, I also had the opportunity to collaborate with several other researchers. Vashist Avadhanula, Karthik Abinav Sankararaman, and Abbas Kazerouni contributed to Chapter 2. Theja Tulabandhula and Prasoon Patidar were a part of Chapter 4. Ciamac Moallemi and Nikhil Bhat were a part of the project which led to Chapter 5 in the thesis. Further, I worked closely with Disha Bhanot, Jackie Baek, Chinmay Jha, and Neal Kaw. I have learned something from each of them and would like to thank them.

My internships at Xerox Research Centre India, Target Corporation, and Facebook Inc. played an important role in shaping my thoughts on the applicability of my research. In fact, two chapters in the thesis stemmed from collaborations formed during these internships. I am thankful to my mentors, colleagues, and co-interns for making these experiences fruitful and memorable.

I consider myself lucky indeed to have been a part of the Operations Research Center (ORC) at MIT. Being filled with many smart yet friendly and warm students, the ORC turned out to

be a great place to pursue graduate studies. Many things in the ORC fell in place because of Laura Rose's and Andrew Carvalho's consistent administrative support. I am thankful to the ORC community for this conducive and intellectually stimulating work environment.

Along with optimization research, baking became a very integral part of my life as a graduate student [1]. Many things fell in place so that I was able to pursue this hobby beyond the confines of my kitchen. Naomi Carton's encouragement and support led me to do a pilot for a cooking class. MIT Mind, Hand and Heart Fund provided generous funds to roll out these classes to all the graduate dorms at MIT. The student governments of Tang, Ashdown, and Sidney-Pacific dorms were extremely welcoming and supportive in this pursuit. Most importantly, the support from friends who came forward and shared their cooking knowledge was invaluable in building a community centered on cooking. For the great joy that these activities brought me, I would like to thank everyone who supported me in this pursuit.

Friends have played a key role in making the past few years sane and fun and I am truly indebted to them. My flatmates Vaishanvi, Vipasha, Priyanka, and Anasuya made for a home away from home. Divya, Somya, Shwetha, Mukund, In Young made me look forward to planning Friday evening activities right from Monday. Our shared love for desserts and their willingness to always be my first taste-testers encouraged me to pursue baking. Many more friends - Monica, Vatsal, Satish, Pritish, Apoorva, Eduardo, Tamar, Phil, Subha, Kaushal, Shilpa, Rim, Anup, Renuka, Sambhav, Shreya, Nikhil, Gowtham, Vashist, Kavya, Pramod, Anuja, Rima, Sara and others added charm to my life. If it were not for Vashist's insistence on considering a PhD in Operations Research or his introducing me to my now-husband, life would have been very different. Amit Mama-Vinita Mami, Bikash Mama-Sujata Mami, and their kids welcomed me to Boston as a part of their family.

I feel deep gratitude towards my family for their unwavering love and support. My parents, Philip and Punam, have inspired me to keep pushing towards goals. My father also played a very active role in establishing the collaboration and planning field visits for my research project on agricultural credit in India. My mother's emphasis on building social relationships has really borne fruit in adjusting to living in a foreign country. My brother, Kshitiz and sister-in-law, Nimisha have always encouraged me in my endeavors - professional or otherwise. With their abundant love, my parents-in-law, Lata and Murty, have been a cheerful addition to my life.

Last, but far from least, I would like to thank my partner-in-many-crimes aka husband,

[1] I could not have done one without the other.

Pradeep. His enthusiasm, patience, love, and support (including IT support) have smoothed a rather uncertain journey as a graduate student.

Contents

1 Introduction **16**

1.1 Motivation . . . 16

1.2 Contribution . . . 17

2 Multi-armed Bandits with Cost Subsidy **21**

2.1 Introduction . . . 21

2.1.1 Problem formulation . . . 23

2.1.2 Related Work . . . 25

2.1.3 Our Contributions . . . 26

2.1.4 Outline . . . 27

2.2 Performance of Existing MAB Algorithms . . . 28

2.3 Lower Bound . . . 29

2.3.1 Proof Overview . . . 31

2.4 Explore-then-commit based algorithm . . . 32

2.5 Performance With Constraints on Costs and Rewards . . . 34

2.5.1 Consistent Cost and Quality . . . 34

2.5.2 Unknown Costs . . . 34

2.6 Numerical Experiments . . . 35

2.6.1 Conclusion and Future Work . . . 36

3 Optimizing Offer Sets in Sub-Linear Time **38**

3.1 Introduction . . . 38

3.1.1 Our Contributions . . . 39

3.1.2 Related Work . . . 41

3.2 Model and Assumptions . . . 42

3.2.1 User and Item Embedding . . . 44

3.2.2 Examples . . . 46
3.3 Algorithm Overview . . . 48
3.4 Our Approach in Detail: Locality-Sensitive Sampling . . . 51
3.4.1 Approximating the Ideal Sampling Distribution via Locality-Sensitive Sampling . . . 51
3.4.2 Aside: LSS Using Locality-Sensitive Hash Functions . . . 53
3.4.3 Putting It All Together . . . 56
3.5 Experiments on Real Data . . . 57
3.5.1 Modeling Diversity in User Behavior . . . 58
3.5.2 Optimization . . . 60
3.6 Conclusion . . . 61

4 Multi-Purchase Behavior: Modeling and Optimization 62
4.1 Introduction . . . 62
4.1.1 Relevant Literature . . . 65
4.2 The BundleMVL Choice Model For Multi-Purchase Behavior . . . 67
4.3 Revenue Maximization: Hardness and Structural Results . . . 71
4.3.1 Hardness of Optimization . . . 72
4.3.2 Structural Properties of the Optimal Solution . . . 72
4.4 Algorithms for Revenue Maximizing Recommendations . . . 73
4.4.1 Binary Search with Efficient Comparisons . . . 74
4.4.2 Optimization with Linear Constraints . . . 77
4.4.3 Benchmark Algorithms . . . 77
4.5 Experiments . . . 79
4.5.1 Suitability of the BundleMVL-2 Model on Real Data . . . 80
4.5.2 Run-times for Computing BundleMVL-2 based Recommendation Sets . . . 84
4.6 Discussion . . . 90
4.7 Conclusion . . . 91

5 Near Optimal A-B Testing 92
5.1 Introduction . . . 92
5.1.1 This Paper . . . 95
5.1.2 Related Literature . . . 97

5.2 Model . . . 100
5.2.1 Setup . . . 100
5.2.2 Optimization Problem . . . 101
5.2.3 Upper Bound, Efficiency, and Loss . . . 102
5.2.4 Problem Interpretation . . . 103
5.3 The Offine Optimization Problem . . . 104
5.3.1 Approximation Algorithm for (P1) . . . 104
5.3.2 Optimal Allocations vs. Randomized Allocations . . . 105
5.4 Sequential Problem . . . 109
5.4.1 Formulation and Surrogate Problem . . . 109
5.4.2 Approximation Guarantee for the Surrogate Problem . . . 111
5.4.3 Dynamic Programming Decomposition . . . 114
5.4.4 State Space Collapse . . . 115
5.5 Variations of the Sequential Problem: A Dynamic Programming Framework . . . 118
5.6 Experiments . . . 122
5.6.1 BCDs, Loss, and Selection Bias . . . 122
5.6.2 Data . . . 123
5.6.3 Algorithms . . . 124
5.6.4 Results . . . 126
5.7 Conclusion . . . 127

6 Conclusion **132**

A Appendix to Chapter 2 **133**
A.1 Technical Lemmas . . . 133
A.2 Proof of Lower Bound . . . 135
A.3 Performance of Algorithms . . . 139
A.4 Algorithm with Unknown and Random Costs . . . 146

B Appendix to Chapter 3 **147**
B.1 Sample Complexity of the Sample Average Approximation . . . 147
B.2 Additional Proofs . . . 148
B.2.1 Proof of Lemma 3.4.2 . . . 148

B.2.2 Proof of Proposition 3.4.3 149

C Appendix to Chapter 4 **151**

C.1 Data Augmentation before MLE for the BundleMVL-K Model 151

C.2 Unconstrained Revenue Optimization under the BundleMVL-2 Model: Hardness Result and Structural Properties of the Optimal Solution 152

C.3 Benchmark Algorithms under the BundleMVL-2 Model 154

C.4 Additional Experiments 158

D Appendix to Chapter 5 **161**

D.1 Derivation of the Optimization Problem 161

D.2 Performance of the Randomized Algorithm 162

D.3 Asymptotic Performance of the Optimal Design 164

D.4 Approximation Guarantee for the Surrogate Problem 169

D.5 Dynamic Programming Formulation 174

D.6 State Space Collapse 175

D.6.1 Proof of Theorem 5.4.7 175

List of Figures

2.1 Cost regret of various algorithms for an instance where the mean reward of the optimal arm is very close to the smallest tolerated reward. CS-TS and CS-UCB incur significant regret. But CS-ETC attains low cost regret. The width of the error bands is two standard deviations based on 50 runs of the simulation. . . . 30

2.2 Performance of algorithms with varying mean reward of the cheaper arm. The length of the error bars correspond to two standard deviations in regret obtained by running the experiment 50 times. 37

3.1 Visual depiction of the approximate sampling scheme. The red curve contains the ideal sampling probabilities $p(\cdot)$, and the blue curve shows how we attempt to approximate it using a step function. In this example, $R = 4$. 52

3.2 Example of sampling of products using Locality-Sensitive Sampling. The sampling probability achieved by the locality sensitive sampling procedure are shown, along with the target sampling distribution and lower bound. 55

3.3 Example of content recommendation by Outbrain. 58

4.1 CDF of the number of products purchased per transaction in the Walmart dataset. 65

4.2 Performance plots. ((a)): Optimality gap of the optimal recommendation set as per the MNL model with the ground truth as the BundleMVL-2 model. ((b)): Time taken to solve for the optimal recommendation set for the unconstrained optimization problem under different models. ((c)): Optimality gap of the optimal recommendation set under the BundleMVL-2 model with the BundleMVL-3 model as the ground truth. ((d)): Time taken to solve the unconstrained optimization problem under different models. Ta Feng-BundleMVL-2 and UCI-BundleMVL-2 run-times overlap. ((e)): Assesing optimality of revenue-ordered heuristic. 83

4.3 Optimality gap and run-time analysis for the unconstrained optimization problem on the Ta Feng dataset. *Fraction of suboptimal solutions:* In ((a)), BinarySearchAO(Efficient), NoisyBinarySearchAO(Efficient), Adxopt2 and Adxopt return no suboptimal solution when number of products $n \leq 60$. In ((b)), revenue-ordered heuristic seems to trail in terms of obtaining optimal solutions. *Optimality gap:* In ((c)), all the algorithms have the median optimality gap as 0. In ((d)), BinarySearchAO(Efficient) and NoisyBinarySearchAO(Efficient) curves overlap and very close to 0. Also, the gaps for revenue-ordered are small in the context of this dataset. *Run-times:* In ((e)) Adxopt, MIP, BinarySearchAO and revenue-ordered curves are close to zero, but they increase a lot for larger sizes. In ((f)) we see that revenue-ordered heuristic is much faster as expected. 86

4.4 Optimality gap and run-time analysis for the optimization problem with capacity constraints on the Ta Feng dataset. ((a)): BinarySearchAO, MIP and Adxopt gets optimal solution in most runs, i.e the plots overlaps near zero value. ((b)) and ((d)): NoisyBinarySearchAO uses the revenue-ordered solution as a lower bound and generates no gain on top of it. The curves overlap completely. ((c)): Adxopt, Adxopt2, MIP and BinarySearchAO plots overlap as the median optimality gap is zero in the small product size regime. Revenue-ordered's gap improves from ((c)) to ((d)) due to an increase in the capacity constraint. . . 89

5.1 Bias-loss trade-off on synthetic Gaussian data for $n = 100$ and varying values of p. 128

5.2 Bias-loss trade-off on synthetic Gaussian data for $n = 1000$ and varying values of p. 129

5.3 Bias-loss trade-off on the Yahoo! dataset for $n = 100$ and varying values of p. . . 130

C.1 Optimality and run-time plots on the UCI dataset in the unconstrained setting. . 159

C.2 Optimality and run-time plots on the UCI dataset in the constrained setting. . . 159

C.3 For the MMC model: (a): run-time vs products, (b) & (c): run-time vs the number of correction sets. 160

List of Tables

2.1 Parameter values 35

3.1 Accuracy of our model of user behavior (Mixed) compared against two single-point benchmarks (Mean and Last). The area under the ROC curve (AUC) and average precision are reported for these models, for a variety choices of the multinomial logit tuning parameters (σ, w) and sampling exponent (α). Results are aggregated over the entire set of test users, replicated 20 times each. 59

3.2 Comparison of our algorithm (LSS) to two common practice benchmarks (Mean, Last) on a recommendation problem for mixed multinomial logit users. Each algorithm's conversion rate is reported, averaged over all test users. For each algorithm, the percentage of test users for which that algorithm achieved the highest conversion is also reported. 60

4.1 Summary of results. 65

4.2 Summary of the datasets used for comparing empirical fit of different models. . . 80

4.3 Log-likelihood values under different models for the Walmart dataset. 81

4.4 Performance analysis of QUBO heuristics: fraction of times the top heursitics gave the best solution 85

4.5 Summary of optimization algorithms and the underlying multi-choice models. . . 86

C.1 Log-likelihood values under different models for six additional datasets. 158

Chapter 1

Introduction

1.1. Motivation

During the last decade, there has been a surge in online platforms for providing a wide variety of services. For many people, daily life has become dependent on these services ranging from purchase and sale of goods, obtaining news, communicating with friends and family, and finding jobs, accommodation, and transportation. In providing their services, the online platforms face an array of challenges that can be mitigated with appropriate modeling and the use of optimization tools. In this thesis, we examine, model, and provide solutions to some of the key challenges.

For any online platform, the first step in building a user base is a smooth and secure onboarding of users onto the platform. This is most commonly achieved by doing a two-factor authentication through a text message (SMS). After user onboarding, text messages continue to play an important role in keeping the user informed about significant activities such as appointment reminders, order confirmations, transaction alerts, and even as a direct marketing line. This makes it important for platforms to be concerned about reliable delivery of their text messages, while not losing sight of the costs incurred in the process.

For many platforms such as those for e-commerce, news, and entertainment, after a user is onboarded, effective personalized recommendations become key to keep the user engaged. These recommendations need to capture the intricacies of the user's preferences. Typically, the few recommended items shown to the users need to be chosen from a much larger set of items that are available on the platform. Moreover, this choice needs to be done in real-time. This necessitates the need for fast and well-performing recommendation engines on online platforms.

Finally, the platforms need a framework to validate the performance of different algorithms (such as those for making recommendations, routing SMSes). Typically, this is done via A-B testing where users arriving on the platform are assigned to one of the multiple treatment groups. Various performance metrics are then compared across the treatment groups to determine the best algorithm. For a fair comparison across the treatment groups, it is important to ensure the distribution of users is similar across the groups. Thus, as the users arrive on the platform, they need to be carefully assigned to each of the treatment groups to maintain balance across these groups measured in terms of the numerous features collected by the platform. Further, appropriate assignment of users can help in controlling the number of users needed to get significant estimates of the effect of treatment.

1.2. Contribution

The work in this thesis is motivated by the above problems of SMS routing, personalized real-time recommendations, and A-B testing for online platforms. The four main chapters of the thesis are outlined below.

Multi-armed Bandits with Cost Subsidy

Sending text messages (SMSes) to users is a common action performed by many online platforms and the expenses from this could range in hundreds of millions of dollars per year. Moreover, for many users, text-based authentication is often one of their first interactions with the platform and thus, the quality of this experience can determine their long-term usage of the platform. For every text message that needs to be sent to a user, the platform needs to choose one among multiple SMS aggregators it has contracts with. The aggregator's quality determines how likely it is that the user will receive the SMS and the price set by the aggregator impacts the cost incurred by the online platform. The platform thus needs to make an aggregator choice while balancing these two metrics of quality and cost.

In this chapter, we model the above SMS routing optimization problem as a novel variant of the multi-armed bandit (MAB) problem, *MAB with cost subsidy*. We show that naive generalizations of existing MAB algorithms like Upper Confidence Bound and Thompson Sampling do not perform well for this problem. We then establish a fundamental lower bound of $\Omega(K^{1/3}T^{2/3})$ on the performance of any online learning algorithm for this problem, highlighting

the hardness of our problem in comparison to the classical MAB problem (where T is the time horizon and K is the number of arms). We also present a simple variant of *explore-then-commit* and establish near-optimal regret bounds for this algorithm. Lastly, we perform extensive numerical simulations to understand the behavior of a suite of algorithms for various instances and recommend a practical guide to employ different algorithms.

Optimizing Offer Sets in Sub-Linear Time

Personalization and recommendations are now accepted as core competencies in just about every online setting, ranging from media platforms to e-commerce to social networks. While the challenge of estimating user preferences has garnered significant attention, the operational problem of using such preferences to construct personalized offer sets to users is still a challenge, particularly in modern settings where a massive number of items and a millisecond response time requirement mean that even enumerating all of the items is impossible. Faced with such settings, existing techniques are either (a) entirely heuristic with no principled justification, or (b) theoretically sound, but simply too slow to work.

Thus motivated, we propose an algorithm for personalized offer set optimization that runs in time sub-linear in the number of items while enjoying a uniform performance guarantee. Our algorithm works for an extremely general class of problems and models of user choice that includes the mixed multinomial logit model as a special case. We achieve a sub-linear runtime by leveraging the dimensionality reduction from learning an accurate latent factor model, along with existing sub-linear time approximate near neighbor algorithms. Our algorithm can be entirely data-driven, relying on samples of the user, where a 'sample' refers to the user interaction data typically collected by firms. We evaluate our approach on a massive content discovery dataset from Outbrain that includes millions of advertisements. Results show that our implementation indeed runs fast and with increased performance relative to existing fast heuristics.

Multi-Purchase Behavior: Modeling and Optimization

In this chapter, we study the problem of modeling the purchase of multiple items and utilizing them to display optimized recommendations, which is a central problem for online e-commerce platforms. Rich personalized modeling of users and fast computation of optimal products to display given these models can lead to significantly higher revenues and simultaneously enhance the end-user experience. We present a parsimonious multi-purchase family of choice models

called the BundleMVL-K family, and develop a binary search based iterative strategy that efficiently computes optimized recommendations for this model. This is one of the first attempts at operationalizing multi-purchase class of choice models.

Further, we characterize structural properties of the optimal solution, which allow one to decide if a product is part of the optimal assortment in constant time, reducing the size of the instance that needs to be solved computationally. We also establish the hardness of computing optimal recommendation sets. We show one of the first quantitative links between modeling multiple purchase behavior and revenue gains. The efficacy of our modeling and optimization techniques compared to competing solutions is shown using several real-world datasets on multiple metrics such as model fitness, expected revenue gains, and run-time reductions. The benefit of taking multiple purchases into account is observed to be 6-8% in relative terms for the Ta Feng and UCI shopping datasets when compared to the MNL model for instances with 1500 products. Additionally, across 8 real-world datasets, the test log-likelihood fits of our models are on average 17% better in relative terms. The simplicity of our models and the iterative nature of our optimization technique allows practitioners to meet stringent computational constraints while increasing their revenues in practical recommendation applications at scale.

Near Optimal AB Testing

A-B Testing has become a ubiquitous tool for many online platforms today. They are used widely not just for deciding visuals of web pages but as a key tool for choosing between different algorithms for a multitude of tasks. In the context of online platforms, A-B testing presents some unique challenges. First, these platforms observe a large number of user covariates and thus would want to achieve balance between the test and control groups on a larger number of features. Second, users arrive in an online fashion. Thus, the allocation needs to be sequential and incorporate uncertainty in the covariates of the un-seen users. Third, because of the large number of A-B tests that are often run in parallel, the number of users available for each A-B test can be limited. This calls for an efficient allocation of users across the test and control groups to get good estimates of the treatment effect with a small number of users.

In this work, we consider the problem of A-B testing in the presence of these challenges, for online platforms. Randomization can be highly inefficient in these settings, and thus we consider the problem of optimally allocating test subjects to either treatment with a view to maximizing the precision of our estimate of the treatment effect. Our main contribution is a

tractable algorithm for this problem in the online setting, where subjects arrive, and must be assigned, sequentially, with covariates drawn from an elliptical distribution with finite second moment. We further characterize the gain in precision afforded by optimized allocations relative to randomized allocations and show that this gain grows large as the number of covariates grows. Our dynamic optimization framework admits several generalizations that incorporate important operational constraints such as the consideration of selection bias, budgets on allocations, and endogenous stopping times. In a set of numerical experiments, we demonstrate that our method simultaneously offers better statistical efficiency and less selection bias than state-of-the-art competing biased coin designs.

Chapter 2

Multi-armed Bandits with Cost Subsidy

2.1. Introduction

In the traditional (stochastic) MAB problem (Robbins (1952)), the learning agent has access to a set of K actions (arms) with unknown, but fixed reward distributions and has to repeatedly select an arm to maximize the cumulative reward. Here, the challenge is designing a policy that balances the tension between acquiring information about arms with little historical observations and exploiting the most rewarding arm based on existing information. The aforementioned exploration-exploitation trade-off has been extensively studied leading to a number of simple but extremely effective algorithms like Upper Confidence Bound (Auer et al. (2002b)) and Thompson Sampling (Thompson (1933), Agrawal and Goyal (2017a)), which have been further generalized and applied in a wide range of application domains including online advertising (Langford and Zhang (2008), Cesa-Bianchi et al. (2014), Oliver and Li (2011)), recommendation systems (Li et al. (2015, 2011), Agrawal et al. (2016)), social networks and crowd sourcing (Anandkumar et al. (2011), Sankararaman et al. (2019), Slivkins and Vaughan (2014)); see Bubeck and Cesa-Bianchi (2012) and Slivkins (2019) for a detailed review. However, most of these approaches cannot be generalized to settings involving multiple metrics (for example reward and cost) when the underlying trade-offs between these metrics are not known a priori.

In many real-world applications of MAB, some of which we will elaborate below, it is common for the agent to incur costs to play an arm, with *high performing arms* costing more. Though, one can model this in the traditional MAB framework by considering cost subtracted from the reward as the modified objective, such a modification is not always meaningful, particularly in settings where the reward and cost associated with an arm represent different quantities (for

example click rate and cost of an ad). In such problems, it is natural for the learning agent to *optimize* for both the metrics, typically trying to avoid incurring exorbitant costs for a marginal increase in cumulative reward. Motivated by the aforementioned scenario, in this paper, we consider a variant of the MAB problem, where the agent is not only concerned about balancing the exploration-exploitation trade-offs to maximize the cumulative reward but also balance the trade-offs associated with multiple objectives that are intrinsic to several practical applications. More specifically, in this work, we study a stylized problem, where to manage costs, the agent is willing to tolerate a small loss from the *highest reward* measured as the reward that could be obtained by the traditional MAB problem in absence of costs. We refer to this problem as MAB problem with a cost subsidy (see Sec 2.1.1 for exact problem formulation), where the subsidy refers to the amount of reward the learning agent is willing to forgo to improve costs. Before we explain our problem and technical contributions in detail, we will elaborate on the applications that motivate this problem.

Intelligent SMS Routing. Many businesses such as banks, delivery services, airlines, hotels, and various online platforms send SMSes (text messages) to their users for a variety of reasons including two-factor authentication, order confirmations, appointment reminders, transaction alerts, and as a direct marketing line (see Twilio and Uber (2020)). These text messages referred to as Application-to-Person (A2P) messages constitute a significant portion of all text messages sent through cellular networks today. In fact, A2P messages are forecasted to be a $86.3 billion business by 2025 (MarketWatch (2020)).

To deliver these messages, businesses typically enlist the support of telecom aggregators, who have private agreements with mobile operators. Each aggregator offers a unique combination of quality, as measured by the fraction of text messages successfully delivered by them and price per message. Surprisingly, it is common for delivery rates of text messages to not be very high (see Canlas et al. (2010), Meng et al. (2007), Zerfos et al. (2006) for QoS analysis in different geographies) and for aggregator's quality to fluctuate with time due to various reasons ranging from network outage to traffic congestion. Therefore, the platform's problem of balancing the tension between inferring aggregator's quality through exploration and exploiting the current *best performing aggregator* to maximize the number of messages delivered to users leads to a standard MAB formulation. However, given the large volume of messages that need to be dispatched, an MAB based solution that focuses exclusively on quality of the aggregator could result in exorbitant spending for the business. A survey of businesses shows that the number of

text messages they are willing to send will have a significant drop if the cost per SMS is increased by a few cents per SMS (Ovum (2017)). Moreover, in many situations platforms have back up communication channels such as email based authentication or notifications via in-app/website features, which though not as effective as a text message in terms of read rate, can be used if guaranteeing the text message delivery proves to be very costly. Therefore, it is natural for businesses to prefer an aggregator with lower costs as long as their quality is comparable to the aggregator with the best quality.

Ad-audience Optimization. We now describe another real-world application in the context of online advertisements. Many advertisers (especially small-to-medium scale businesses) have increasingly embraced the notion of *auto-targeting* where they let the advertising platform identify a *high-quality* audience group (*e.g.,* Koningstein (2006), Amazon (2019), Facebook (2016), Google (2014)). To enable this, the platform explores the audience-space to identify *cheaper* opportunities that also give high click-through-rate (ctr) and conversion rate. Here, it is possible for different audience groups to have different yields i.e. quality (CTR/conversion rate) for a specific ad. However, it may require vastly different bids to reach different audiences due to auction overlap with other ad campaigns with smaller audience targeting. Thus, the algorithm is faced with a similar trade-off; as long as a particular audience-group gives a high-yield, the goal is to find the cheapest one.

We now present a novel formulation of a multi-armed bandit problem that captures key features of these applications, where our goal is to develop a cost sensitive MAB algorithm that balances both the exploration-exploitation trade-offs as well as the tension between conflicting metrics in a multi-objective setting.

2.1.1. Problem formulation

To formally state our problem, given an instance I, in every round $t \in [T]$ the agent chooses an arm $i \in [K]$ and realizes a reward r_t, sampled independently from a fixed, but unknown distribution $\mathcal{F}_i$ with mean μ_i (or μ_i^I) and incurs a cost c_i (or c_i^I), which is known a priori. Here, in order to manage costs, we allow the agent to be agnostic between arms, whose expected reward is greater than $1-\alpha$ fraction of the highest expected reward, for a fixed and known value of α, which we refer to as the *subsidy factor*. The agent's objective is to learn and pull the cheapest arm among these *high* quality arms as frequently as possible.

More specifically, let m_* denote the arm with highest expected mean, *i.e.,* $m_* = \text{argmax}\ \mu_i$,

and $\mathcal{C}_*$ be the set of arms whose expected reward is within $1-\alpha$ factor of the highest expected reward, *i.e.,* $\mathcal{C}_* = \{i \mid \mu_i \geq (1-\alpha)\mu_{m_*}\}$. We refer to the quantity $(1-\alpha)\mu_{m_*}$ as the *smallest tolerated reward*. Without loss of generality, we assume the reward distribution has support $[0,1]$. The goal of the agent is to design a policy (algorithm) π that will learn the cheapest arm whose expected reward is at least as large as the smallest tolerated reward. In other words, the agent needs to learn the identity and simultaneously maximize the number of plays of arm $i_* = \text{argmin}\,\{c_i \mid i \in \mathcal{C}_*\}$. Since in the SMS application, the reward is the quality of the chosen aggregator, we will use the terms reward and quality interchangeably.

To measure the performance of any policy π, we propose two notions of regret - quality and cost regret, with the agent's goal being minimizing both of them:

$$\begin{aligned}
&\mathsf{Quality_Reg}_\pi(T, \alpha, \boldsymbol{\mu}, \boldsymbol{c}) \\
&\qquad = \mathbb{E}\left[\sum_{t=1}^{T} \max\{(1-\alpha)\mu_{m_*} - \mu_{\pi_t}, 0\}\right], \\
&\mathsf{Cost_Reg}_\pi(T, \alpha, \boldsymbol{\mu}, \boldsymbol{c}) \\
&\qquad = \mathbb{E}\left[\sum_{t=1}^{T} \max\{c_{\pi_t} - c_{i_*}, 0\}\right],
\end{aligned} \tag{2.1}$$

where $\boldsymbol{c} = (c_1, \cdots c_K), \boldsymbol{\mu} = (\mu_1, \cdots \mu_k)$ and the expectation is over the randomness in policy π. Equivalently, the cost and quality regret of policy π on an instance I of the problem is denoted as $\mathsf{Quality\_Reg}_\pi(T, \alpha, I)$ and $\mathsf{Cost\_Reg}_\pi(T, \alpha, I)$ where the instance is defined by the distributions of the reward and cost of each arm. The objective then is to design a policy that simultaneously minimizes both the cost and quality regret for all possible choices of $\boldsymbol{\mu}$ and $\boldsymbol{c}$ (equivalently all instances I).

Remark 2.1.1 (Choice of objective function)**.** Note that a parametrized linear combination of reward and cost metrics, i.e. $\mu - \lambda c$ for an appropriately chosen λ is a popular approach to balance cost-reward trade-off. However, the setting considered in this paper is not equivalent to this approach. In particular, for any specified subsidy factor α, the value λ required in the linear objective function, for i_* to be the optimal arm would depend on the cost and reward distributions of the arms. Therefore, using a single value of λ and relying on standard MAB algorithms would not lead to the desired outcome for our problem. Further, from an application stand-point there are two important considerations. First, in a real-world system we need an explicit control over the parameter α that is not instance-dependent to understand and defend

the trade-off between the various objectives. Second, for the intelligent SMS routing application discussed earlier, different sets of aggregators operate in different regions. Thus, separate λ values would need to be configured for each region, making the process cumbersome.

2.1.2. Related Work

Our problem is closely related to the MAB with multiple objectives line of work, which has attracted considerable attention in recent times. The existing literature on multi-objective MAB can be broadly classified into three different categories.

Bandits with Knapsacks (BwK). Bandits with knapsacks (BwK), introduced in the seminal work of Badanidiyuru et al. (2018) is a general framework that considers the standard MAB problem under the presence of additional budget/resource constraints. The BwK problem encapsulates a large number of constrained bandit problems that naturally arise in many application domains including dynamic pricing, auction bidding, routing and scheduling ((see Tran-Thanh et al. (2012), Agrawal and Devanur (2014), Immorlica et al. (2019)). In this formulation, the agent has access to a set of d finite resources and K arms, each associated with a reward distribution. Upon playing arm a at time t, the agent realizes a reward of r_t and incurs a penalty of $c_t^{(i)}$ for resource i, all drawn from a fixed, but unknown distribution corresponding to the arm. The objective of the agent is to maximize the cumulative reward before one of the resources is completely depleted. Although appealing in many applications, BwK formulation requires *hard* constraint on resources (cost in our setting) and hence, cannot be easily generalized to our problem. In particular, in the cost subsidized MAB problem, the equivalent budget limits depend on the problem instance and therefore cannot be determined a priori.

Pareto Optimality and Composite Objective. The second formulation is focused on identifying Pareto optimal alternatives and uniformly choosing among these options (see Drugan and Nowe (2013), Yahyaa et al. (2014), Paria et al. (2018), Yahyaa and Manderick (2015)). These approaches do not apply to our problem, since some of the Pareto alternatives could have extreme values for one of the metrics, for example having very low cost and low quality or extremely high cost and quality, making them undesirable for the applications discussed earlier. Closely related to this line of work is the set of works that focus on a composite objective by appropriately weighting the different metrics (see Paria et al. (2018), Yahyaa and Manderick (2015)). Such formulations also do not immediately apply for our problem, since in the SMS and ad applications discussed earlier, it is not acceptable to drop the quality beyond the allowed

level irrespective of the cost savings we could obtain. Furthermore, in the SMS application, the trade-offs between quality and costs could vary from region to region, making it hard to identify a good set of weights for the composite objective (see Remark 2.1.1).

Conservative Bandits and Bandits with Safety Constraints. Two other lines of work that are recently receiving increased attention, particularly from practitioners are *bandits with safety constraints* (*e.g.,* Daulton et al. (2019), Amani et al. (2020), Galichet et al. (2013)) and *conservative bandits* (*e.g.,* Wu et al. (2016), Kazerouni et al. (2017)). In both these formulation, the algorithm chooses one of the arms and receives a reward and a cost associated with it. The goal of the algorithms is to maximize the total reward obtained while ensuring that either the chosen arm is within a pre-specified threshold (when costs of arms are unknown a priori) or reward of the arm is at least a specified fraction of a known benchmark arm. Neither of these models exactly capture the requirements of our applications: a) we do not have a hard constraint on the acceptable cost of a pulled arm. In particular, choosing low quality aggregators to avoid high costs (even for a few rounds) could be disastrous since it leads to bad user experience on the platform and eventual churn. b) the equivalent benchmark arm in our case i.e. the arm with the highest mean reward is not known a priori.

Best Arm Identification. Apart from the closely related works mentioned above, our problem of identifying the *cheapest* arm whose expected reward is within an acceptable margin from the highest reward can be formulated as a stylized version of the *best-arm identification problem* (Katz-Samuels and Scott (2019), Jamieson and Nowak (2014), Chen et al. (2014), Cao et al. (2015), Chen et al. (2016)). However, in many settings and particularly applications discussed earlier, the agent's objective is optimizing cumulative reward and not just identifying the *best arm*.

2.1.3. Our Contributions

Novel Problem Formulation. In this work, we propose a stylized model, *MAB with a cost subsidy* and introduce new performance metrics that uniquely capture the salient features of many real world online learning problems involving multiple objectives. For this problem, we first show that naive generalization of popular algorithms like Upper Confidence Bound (UCB) and Thompson Sampling (TS) could lead to poor performance on the metrics. In particular, we show that the naive generalization of TS for this problem would lead to a linear cost regret for some problem instances.

Lower Bound. We establish a fundamental limit on the performance of *any online algorithm* for our problem. More specifically, we show that any online learning algorithm will incur a regret of $\Omega(K^{1/3}T^{2/3})$ on either the cost or the quality metric (refer to (2.1)), further establishing the *hardness of our problem* relative to the standard MAB problem, for which it is possible to design algorithms that achieve worst case regret bound of $\tilde{O}(\sqrt{KT})$. We introduce a novel reduction technique to derive the above lower bound, which is of independent interest.

Cost Subsidized Explore-Then-Commit. We present a simple algorithm, based on the *explore-then-commit* (ETC) principle and show that it achieves near-optimal performance guarantees. In particular, we establish that our algorithm achieves a worst-case bound of $O(K^{1/3}T^{2/3}\sqrt{\log T})$ for both cost and quality regret. A key challenge in generalizing the ETC algorithm for this problem arises from having to balance between two asymmetric objectives. We also discuss generalizations of the algorithm for settings where cost of the arms is not known a priori. Furthermore, we consider a special scenario of bounded costs, where naive generalizations of TS and UCB work reasonably well and establish worst case regret bounds.

Numerical Simulation. Lastly, we perform extensive simulations to understand various regimes of the problem parameters and compare different algorithms. More specifically, we consider scenarios where naive generalizations of UCB and TS, which have been adapted in real life implementations (see Daulton et al. (2019)) perform well and settings where they perform poorly, which should be of interest to practitioners.

2.1.4. Outline

The rest of this paper is structured as follows. In Section 2.2, we show that the naive generalization to TS or UCB algorithms perform poorly and in Section 2.3, we establish lower bounds on performance of any algorithm for MAB with cost subsidy problem. In Section 2.4, we present a variation of the ETC algorithm, and show that it achieves a near-optimal regret bound of $\tilde{O}(K^{1/3}T^{2/3})$ for both the metrics. In section 2.5, we show that with additional assumptions it is possible to show improved performance bounds for naive generalization of existing algorithms. Finally, in section 2.6 we perform numerical simulations to explore various regimes of the instance-space.

2.2. Performance of Existing MAB Algorithms

In this section, we consider a natural extension of two popular MAB algorithms, TS and UCB for our problem and show that such adaptations perform poorly. This highlights the challenges involved in developing good algorithms for the MAB problem with cost subsidy. In particular, we establish theoretically that for some problem instances the TS variant incurs a linear cost regret and observe similar performance for the UCB variant empirically. Our key focus on TS in this section is primarily motivated by the superior performance that have been observed over a stream of recent papers in the context of TS versus more traditional approaches such as UCB (see Scott (2010), Oliver and Li (2011), May et al. (2012), Agrawal et al. (2017)).

We present the details of TS and UCB adaptations in Algorithm 1, which we will refer to as Cost-Subsidized TS(CS-TS) and Cost-Subsidized UCB(CS-UCB) respectively. These extensions are inspired by Daulton et al. (2019), which demonstrates empirical efficacy on a related (but different) problem. Briefly, in the CS-TS(CS-UCB) variation, we follow the standard TS (UCB) algorithm and obtain a quality score which is a sample from the posterior distribution (upper confidence bound) for each arm. We then construct a feasible set of arms consisting of arms whose quality scores are greater than $1-\alpha$ fraction of the highest quality score. Finally, we pull the cheapest arm among the feasible set of arms.

Algorithm 1 Cost Subsidized TS and UCB Algorithms

Require: T, K, prior distribution for mean rewards of all arms $\{\nu_i\}_{i=1}^K$, reward likelihood function $\{L_i\}_{i=1}^K$
  $T_i(1) = 0 \ \forall i \in [K]$
  for $t \in [K]$ **do**
    $I_t = t$
    Play arm I_t and observe reward r_t
    $T_i(t+1) = T_i(t) + \mathbf{1}\{I_t = i\} \ \forall i \in [K]$
  for $t \in [K+1, T]$ **do**
    for $i \in [K]$ **do**
      $\hat{\mu}_i(t) \leftarrow \left(\sum_{\tau=1}^{t-1} r_\tau \mathbb{1}\{I_\tau = i\}\right) / T_i(t)$
      $\beta_i(t) \leftarrow \sqrt{(2\log T)/T_i(t)}$
      UCB: $\mu_i^{score}(t) \leftarrow \min\{\hat{\mu}_i(t) + \beta_i(t), 1\}$
      TS: Sample $\mu_i^{score}(t)$ from the posterior distribution of arm i,
      $\nu_i(\cdot \,|\{r_s\}_{s\in\{1,2,\cdots t-1\} \text{ s.t. } I_s=i}, L_i)$
    $m_t = \arg\max_i \mu_i^{score}(t)$
    $Feas(t) = \{i : \mu_i^{score}(t) - (1-\alpha)\mu_{m_t}^{score} \geq 0\}$
    $I_t = \arg\min_{i \in Feas(t)} c_i$
    Play arm I_t and observe reward r_t
    $T_i(t+1) = T_i(t) + \mathbf{1}\{I_t = i\} \ \forall i \in [K]$
  return Arm I_t to be pulled in each round $t \in [T]$

We will now show that CS-TS with Gaussian priors and posteriors (i.e. Gaussian distribution with mean $\hat{\mu}_i(t)$ and variance $1/T_i(t)$) described in Algorithm 1 incurs a linear cost regret in the worst case. More precisely, we prove the following result.

Theorem 2.2.1. For any given K, α, T there exists an instance ϕ of problem such that $\mathsf{Quality\_Reg}_{\mathsf{CS-TS}} + \mathsf{Cost\_Reg}_{\mathsf{CS-TS}}(T, \alpha, \phi)$ is $\Omega(T)$.

Proof Sketch. The proof closely follows the lower bound argument in Agrawal and Goyal (2017b). We briefly describe the intuition behind the result. Consider a scenario where the highest reward arm is expensive arm while all other arms are cheap and have rewards marginally above the smallest tolerated reward. In the traditional MAB problem, the anti-concentration property of the Gaussian distribution (see Agrawal and Goyal (2017b)) ensures samples from *good arm* would be large enough with sufficient frequency, ensuring appropriate exploration and good performance. However, in our problem, the anti-concentration property would result in playing the expensive arm too often since the difference in the mean qualities is small, incurring a linear cost regret while achieving zero quality regret. A complete proof of the theorem is provided in Appendix A.3. ■

The poor performance of the algorithm is not limited only to the above instance and usage of Gaussian prior. More generally, the CS-TS and CS-UCB algorithms seem to perform poorly whenever the mean reward of the optimal arm is very close to the smallest tolerated reward. We illustrate this through another empirical example. Consider the following instance with two arms each having Bernoulli rewards and $T = 10,000$. The costs of the two arms are $c_1 = 0$ and $c_2 = 1$. The expected qualities are $\mu_1 = 0.5(1-\alpha) + 1/\sqrt{T}, \mu_2 = 0.5$ with $\alpha = 0.1$. The prior of the mean reward of both the arms is a Beta(1,1) distribution. Here, the quality regret will be zero irrespective of which arm is played. But both CS-TS and CS-UCB incur significant cost regret as shown in Figure 2.1. (In the figure, we also plot the performance of the key algorithm we propose in the paper (Algorithm 2) and note that it has much superior performance as compared to CS-TS and CS-UCB.)

2.3. Lower Bound

In this section, we establish that any policy must incur a regret of $\Omega(K^{1/3}T^{2/3})$ on at least one of the regret metrics. More precisely, we prove the following result.

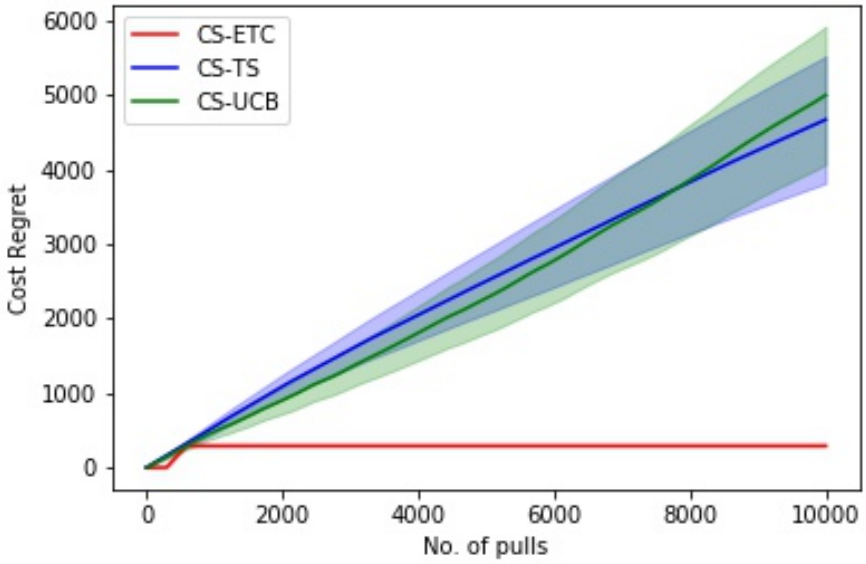


Figure 2.1: Cost regret of various algorithms for an instance where the mean reward of the optimal arm is very close to the smallest tolerated reward. CS-TS and CS-UCB incur significant regret. But CS-ETC attains low cost regret. The width of the error bands is two standard deviations based on 50 runs of the simulation.

Theorem 2.3.1. For any given α, K, T and (possibly randomized) policy π, there exists an instance ϕ of problem (2.1) with $K+1$ arms such that $\mathsf{Quality\_Reg}_\pi(T, \alpha, \phi) + \mathsf{Cost\_Reg}_\pi(T, \alpha, \phi)$ is $\Omega\left((1-\alpha)^2 K^{\frac{1}{3}} T^{\frac{2}{3}}\right)$ when $0 \leq \alpha \leq 1$ and $1 \leq K \leq T$.

2.3.1. Proof Overview

We consider the following families of instances to establish the lower bound. More specifically, we first prove the result for $\theta = 0$ and then establish a reduction for $\theta = \alpha$ to the special case of $\theta = 0$.

Definition 2.3.2 (Family of instances $\Phi_{\theta,p,\epsilon}$)**.** Define a family of instances $\Phi_{\theta,p,\epsilon}$ consisting of instances $\Phi^0_{\theta,p,\epsilon}, \Phi^1_{\theta,p,\epsilon}, \cdots \Phi^K_{\theta,p,\epsilon}$ each with $K+1$ Bernoulli arms indexed by $0, 1, \cdots, K$. For the instance $\Phi^0_{\theta,p,\epsilon}$, the costs and mean reward of the j-th arm are

$$c_j^{\Phi^0_{\theta,p,\epsilon}} = \begin{cases} 0 & j = 0 \\ 1 & j \neq 0 \end{cases}, \qquad \mu_j^{\Phi^0_{\theta,p,\epsilon}} = \begin{cases} p & j = 0 \\ \frac{p}{1-\theta} & j \neq 0 \end{cases}. \text{ for } 0 \leq j \leq K. \text{ For the instance } \Phi^a_{\theta,p,\epsilon} \text{ with}$$

$1 \leq a \leq K$, the costs and mean rewards of the j-th arm are

$$c_j^{\Phi^a_{\theta,p,\epsilon}} = \begin{cases} 0 & j = 0 \\ 1 & j \neq 0 \end{cases}, \qquad \mu_j^{\Phi^a_{\theta,p,\epsilon}} = \begin{cases} p & j = 0 \\ \frac{p+\epsilon}{1-\theta} & j = a \\ \frac{p}{1-\theta} & \text{otherwise} \end{cases}.$$

for $0 \leq j \leq K$, where $0 \leq \theta < 1, 0 < p \leq 1/2, \epsilon > 0$ and $(p+\epsilon)/(1-\theta) < 1$.

Lemma 2.3.3. For any given p, K, T and any (possibly randomized) policy π, there exists an instance ϕ (from the family $\Phi_{0,p,\epsilon}$) such that $\mathsf{Quality\_Reg}_\pi(T, 0, \phi) + \mathsf{Cost\_Reg}_\pi(T, 0, \phi)$ is $\Omega\left(pK^{\frac{1}{3}} T^{\frac{2}{3}}\right)$ when $0 < p \leq 1/2$ and $1 \leq K \leq T$.

Lemma 2.3.3 establishes that when $\alpha = 0$, any policy must incur a regret of $\Omega(K^{1/3}T^{2/3})$ on an instance from the family $\Phi_{0,p,\epsilon}$. To prove Lemma 2.3.3, we argue that any online learning algorithm will not be able to differentiate the instance $\Phi^0_{0,p,\epsilon}$ from the instance $\Phi^a_{0,p,\epsilon}$ for $1 \leq a \leq K$ and therefore, must either incur a high cost regret if the algorithm does not select 0^{th} arm frequently or high quality regret if the algorithm selects 0^{th} arm frequently. More specifically, any online algorithm would require $O(1/\epsilon^2)$ samples or rounds to distinguish instance $\Phi^0_{0,p,\epsilon}$ from instance $\Phi^a_{0,p,\epsilon}$ for $1 \leq a \leq K$. Hence, any policy π can avoid high quality regret by *exploring sufficiently* for $O(1/\epsilon^2)$ rounds, incurring a cost regret of $O(1/\epsilon^2)$ or incur zero cost regret at the expense of $O(T\epsilon)$ regret on the reward metric. This suggests a trade-off between $1/\epsilon^2$ and $T\epsilon$, which are of the same magnitude at $\epsilon = T^{-1/3}$ resulting in the aforementioned lower bound.

The complete proof generalizes techniques from the standard MAB lower bound proof and is provided in Appendix A.2. ■

Now, we generalize the above result for $\alpha = 0$ to any α for $0 \leq \alpha \leq 1$. The main idea in our reduction is to show that if there exists an algorithm π_α for $\alpha > 0$ such that $\mathsf{Quality\_Reg}_\pi(T, \alpha, \phi) + \mathsf{Cost\_Reg}_\pi(T, \alpha, \phi)$ is $o(K^{1/3}T^{2/3})$ on every instance in the family $\Phi_{\alpha,p,\epsilon}$, then we can use π_α as a subroutine to construct an algorithm π for problem (2.1) such that $\mathsf{Quality\_Reg}_\pi(T, 0, \phi) + \mathsf{Cost\_Reg}_\pi(T, 0, \phi)$ is $o(K^{1/3}T^{2/3})$ on every instance in $\Phi_{0,p,\epsilon}$, thus contradicting the lower bound of Lemma 2.3.3. This will prove Theorem 2.3.1 by contradiction. In order to construct the aforementioned sub-routine, we leverage techniques from *Bernoulli factory* (Keane and O'Brien (1994), Huber (2013)) to generate a sample from a Bernoulli random variable with parameter $\mu/(1-\alpha)$ using samples from a Bernoulli random variable with parameter μ, for any $0 < \mu < 1 - \alpha < 1$. We provide the exact sub-routine and complete proof in Appendix A.2.

2.4. Explore-then-commit based algorithm

We propose an explore-then-commit algorithm, named Cost-Subsidized Explore-Then-Commit (CS-ETC), to have better worst case performance guarantees as compared to the extensions of the TS and UCB algorithms. As the name suggests, first this algorithm plays each arm for a specified number of rounds. After sufficient exploration, the algorithm continues in a UCB-like fashion. In every round, based on the upper and lower confidence bounds on the reward of each arm, a feasible set of arms is constructed as an estimate of all arms having mean reward greater than the smallest tolerated reward. The lowest cost arm in this feasible set is then pulled. This is detailed in Algorithm 2. The key question that arises in this algorithm is how many exploration rounds are needed before exploitation can begin. We establish that $O\left((T/K)^{2/3}\right)$ rounds are sufficient for exploration in the following result (proof in Appendix A.3).

Algorithm 2 Cost-Subsidized Explore-Then-Commit

Require: K, T, no. of exploration pulls per arm τ
  $T_i(1) = 0 \ \forall i \in [K]$
  Pure exploration phase:
  for $t \in [1, K\tau]$ **do**
    $I_t = t \mod K$
    Pull arm I_t to obtain reward r_t
    $T_i(t+1) = T_i(t) + \mathbf{1}\{I_t = i\} \ \forall i \in [K]$
  UCB phase:
  for $t \in [K\tau + 1, T]$ **do**
    $\hat{\mu}_i(t) \leftarrow \left(\sum_{\tau=1}^{t-1} r_\tau \mathbb{1}\{I_\tau = i\}\right) / T_i(t) \ \forall i \in [K]$
    $\beta_i(t) \leftarrow \sqrt{(2 \log T) / T_i(t)} \ \forall i \in [K]$
    $\mu_i^{\mathsf{UCB}}(t) \leftarrow \min\{\hat{\mu}_i(t) + \beta_i(t), 1\} \ \forall i \in [K]$
    $\mu_i^{\mathsf{LCB}}(t) \leftarrow \max\{\hat{\mu}_i(t) - \beta_i(t), 0\} \ \forall i \in [K]$
    $m_t = \arg\max_i \mu_i^{\mathsf{LCB}}(t)$
    $Feas(t) = \{i : \mu_i^{\mathsf{UCB}}(t) \geq (1-\alpha)\mu_{m_t}^{\mathsf{LCB}}(t)\}$
    $I_t = \arg\min_{i \in Feas(t)} c_i$
    Pull arm I_t to obtain reward r_t
    $T_i(t+1) = T_i(t) + \mathbf{1}\{I_t = i\} \ \forall i \in [K]$
  return Arm I_t to be pulled in each round $t \in [T]$

Theorem 2.4.1. For an instance ϕ with K arms, when the number of exploration pulls of each arm $\tau = (T/K)^{2/3}$, then the sum of cost and quality regret incurred by CS-ETC(Algorithm 2) on any instance ϕ i.e. $\mathsf{Quality\_Reg}_{CS-ETC}(T, \alpha, \phi) + \mathsf{Cost\_Reg}_{CS-ETC}(T, \alpha, \phi)$ is $O(K^{1/3} T^{2/3} \sqrt{\log T})$.

The key reason that sufficient exploration is needed for our problem is that there can be arms with mean rewards very close to each other but significantly different costs. If cost regret were not of concern, then playing either arm would have led to satisfactory performance by giving low quality regret. But the need for performing well on both cost and quality regrets necessitates differentiating between the two arms and finding the one with the cheapest cost among the arms with mean reward above the smallest tolerated reward.

The regret guarantee mainly stems from the exploration phase of the algorithm. In fact, an algorithm which estimates the optimal arm only once after the exploration phase and pulls that arm for the remaining time will have the same regret upper bound as CS-ETC. But we empirically observed that the non-asymptotic performance of this algorithm is worse as compared to Algorithm 2.

2.5. Performance With Constraints on Costs and Rewards

In this section, we present some extensions of the previous results.

2.5.1. Consistent Cost and Quality

The lower bound result in Theorem 2.3.1 is motivated by an extreme instance where arms with very similar mean rewards have very different costs. This raises the following question - can better performing algorithms be obtained if the rewards and costs are *consistent* with each other? We show that this is indeed the case. Motivated by the instance which led to the worst case performance, we consider a constraint which gives an upper bound on the difference in costs of every pair of arms by a multiple of the difference in the qualities of these arms. Under this constraint, CS-UCB has good performance as per the following result with the proof in Appendix A.3.

Theorem 2.5.1. If for an instance ϕ with K arms, $|c_i - c_j| \leq \delta|\mu_i - \mu_j|$ $\forall i, j \in [K]$ and any (possibly unknown) $\delta > 0$, then $\mathsf{Quality\_Reg}_{CS-UCB}(T, \alpha, \phi) + \mathsf{Cost\_Reg}_{CS-UCB}(T, \alpha, \phi)$ is $O((1+\delta)\sqrt{KT \log T})$.

Note that, in general, δ can be unknown. Hence, even with the above assumption on consistency of cost and quality, a priori any algorithm cannot get a bound on the quality difference between arms, only by virtue of knowing their costs.

2.5.2. Unknown Costs

In some applications, it is possible that the costs of the arms are also unknown and random. Hence, in addition to the mean reward, the mean costs also need to be estimated. Without loss of generality, we assume that the distribution of the random cost of each arm has support [0,1]. Not knowing the cost of the arm, does not fundamentally change the regret minimization problem we have discussed in the above sections. Clearly, the lower bound result is still valid. Algorithm 2 can be generalized to the unknown costs setting with a minor modification in the UCB phase of the algorithm. The modified UCB phase is described in Algorithm 7 in Appendix A.4. In this algorithm, we maintain confidence bounds on the costs of each arm. Instead of picking the arm with the lowest cost among all feasible arms, the algorithm now picks the arm with the lowest lower confidence bound on cost. Theorem 2.4.1 holds for this modified algorithm also.

Parameter	Value
Mean reward of arm 1 (μ_1)	0.5
Mean reward of arm 2 (μ_2)	0.3-0.6
Cost of arm 1 (c_1)	1
Cost of arm 2 (c_2)	0
Subsidy factor (α)	0.1
Time horizon (T)	5000

Table 2.1: Parameter values

Similarly, when costs and quality are consistent as described in Section 2.5.1, the CS-UCB algorithm can be modified to pick the arm with the lowest lower confidence bound on cost and Theorem 2.5.1 holds.

2.6. Numerical Experiments

In the previous sections, we have shown theoretical results on the worst case performance of different algorithms for (2.1). Now, we illustrate the empirical performance of these algorithms. We shed light on which algorithm performs better in what regime of parameter values. The key quantity which differentiates the performance of different algorithms is how close the mean rewards of different arms are to each other. We consider a setting with two Bernoulli arms and vary the mean reward of one arm (the cheaper arm) while keeping the other quantities (reward distribution of the other arm and costs of both arms) fixed. The values of these parameters are described in Table 2.1. The reward in each round follows a Bernoulli distribution whereas the cost is a known fixed value. The cost and quality regret at time T of the different algorithms are plotted in Figure 2.2.

We observe that the performance of the CS-TS and CS-UCB are close to each other for the entire range of mean reward values. To compare the performance of these algorithms with CS-ETC, we focus on how close the mean reward of the lower mean reward arm is to the smallest tolerated reward. When $\mu_2 \leq 0.5$ ($\mu_2 > 0.5$), the lowest tolerated reward is 0.45 ($0.9\mu_2$). In terms of quality regret, when μ_2 is much smaller than 0.45, CS-TS and CS-UCB perform much better than CS-ETC. This is because the number of exploration rounds in the CS-ETC algorithm is fixed (independent of the difference in mean rewards of the two arms) leading to higher quality regret when μ_2 is much smaller than 0.45. On the other hand, because of the large difference in μ_2 and 0.45, CS-TS and CS-UCB algorithms are easily able to find the optimal arm and incur low quality regret. The cost regret of all algorithms is 0 because the optimal arm is the

expensive arm.

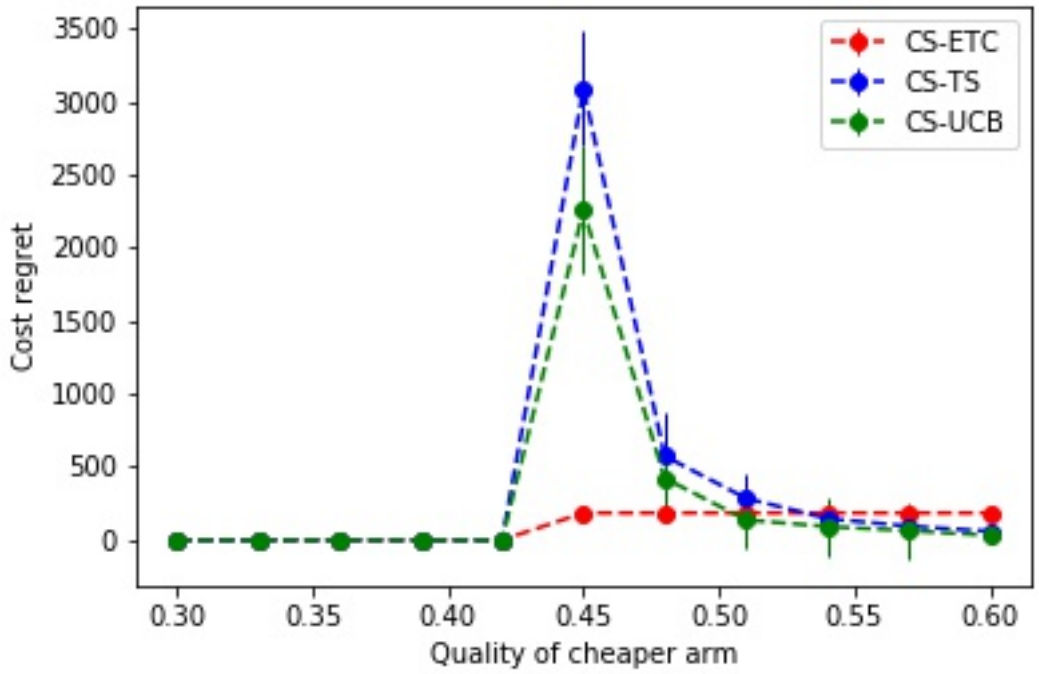
When μ_2 is close to (and less than) 0.45, CS-TS and CS-UCB incur much higher cost regret as compared to CS-ETC. This is in line with the intuition established in Section 2.2. Here, CS-TS and CS-UCB are unable to effectively conclude that the second (cheaper) arm is optimal. Thus, they end up pulling the first (expensive) arm many times leading to high cost regret. On the other hand, CS-ETC, after the exploration rounds is able to correctly identify the second arm as the optimal arm.

Thus, we recommend using the CS-TS/CS-UCB algorithm when the mean rewards of arms are well differentiated and CS-ETC when the mean rewards are close to one another (as is often the case in the SMS application). This is in line with the notion that algorithms which perform well in the worst case might not have good performance for an average case.

2.6.1. Conclusion and Future Work

In this paper, we have proposed a new variant of the MAB problem which factors costs associated with playing an arm and introduces new metrics that uniquely capture the features of multiple real world applications. We argue about the *hardness* of this problem by establishing fundamental limits on performance of any online algorithm and also demonstrating that traditional MAB algorithms perform poorly from a both theoretical and empirical standpoint. We present a simple near-optimal algorithm and through numerical simulations, we prescribe ideal algorithmic choice for different problem regimes.

An important question that naturally arises from this work is developing an algorithm for the adversarial variant of the MAB with cost subsidy problem. In particular, it is not immediately clear if EXP3 (Auer et al. (2002a)) family of algorithms, that are popular for non-stochastic MAB problem can be generalized to setting where the reward distribution is not stationary.



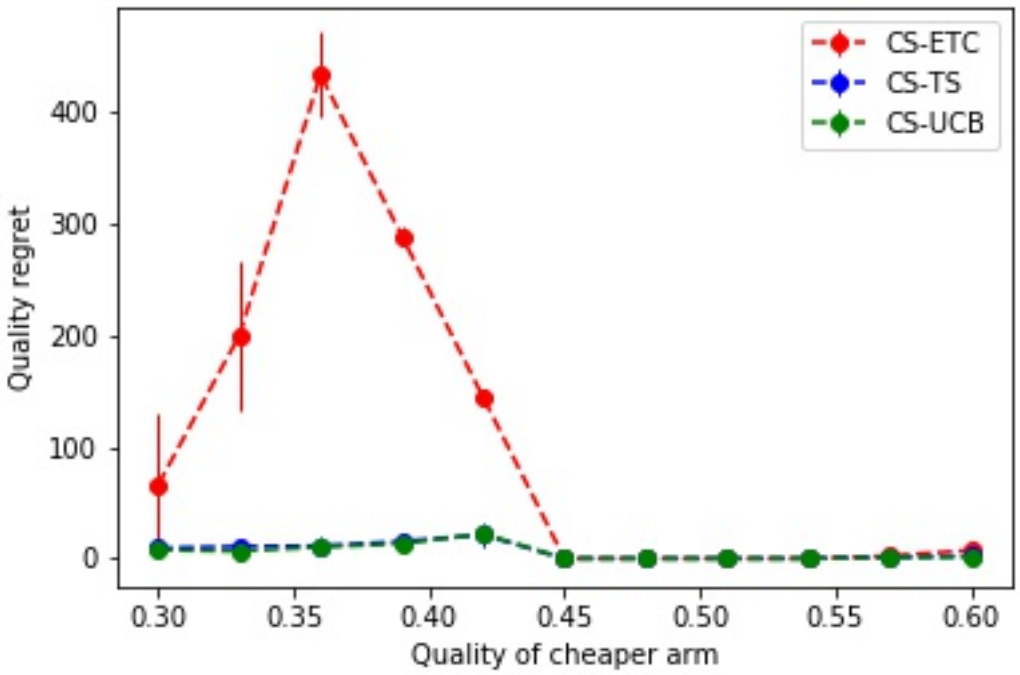


Figure 2.2: Performance of algorithms with varying mean reward of the cheaper arm. The length of the error bars correspond to two standard deviations in regret obtained by running the experiment 50 times.

Chapter 3

Optimizing Offer Sets in Sub-Linear Time

3.1. Introduction

A common problem in modern web-services revolves around constructing personalized offer sets (or assortments) for users with the goal of optimizing some objective function related to each specific user's experience with the service. Recommendation problems (such as those faced by services like Netflix) represent a canonical version of such a problem. Assortment optimization problems (as faced by online retailers) are another common example. Now, in practice, rigorous latency constraints are an important consideration when building algorithms to construct optimized offer sets. For example, a large-scale study (Akamai (2017)) recently showed that a 100 millisecond delay in loading page content can result in a decrease in conversion (i.e. consumption, purchase, etc.) of up to 7%. When taken together with the vast size of the product universe (which can run to the tens of millions), such constraints place severe limitations on what a *real-time* algorithm for constructing an optimized offer can do in practice. As such, practically implementable algorithms for real-time offer set optimization at scale should ideally exhibit runtimes *sub-linear* in the size of the universe of potential products.

The dynamic nature of the product universe limits the use of pre-computation. Moreover, as will be evident later, distributional models of the user further restrict our ability to exhaustively pre-compute optimal offer sets. As a result, the dominant approach today to building optimized offer sets in real-time relies on the design of so-called approximate nearest neighbor (ANN) algorithms. Succinctly, such algorithms leverage metric-space representations of users and

products wherein the distance between a user and product is inversely related to how attractive the product is to the user. Whereas such a metric space representation of products may be constructed offline, the real-time problem then consists of identifying a point in the metric space that corresponds to the user, and finding the k products in the metric space that are nearest to that user in time sub-linear in the number of products. This problem is well-solved both theoretically and practically.

Now an economically grounded approach to constructing an optimal offer-set would typically rely on modeling the user's utility for various products. An offer set constructed assuming that a user made choices to maximize utility would then seek not just to pick products that are likely interesting to the user, but would further seek to account for substitution and complementarity effects. There is by now a vast literature dedicated to the estimation of models of choice as well as the associated assortment optimization problems given such models. The assortment optimization algorithms developed in this context are, while typically efficient, not sub-linear. On the other hand, the ANN paradigm while sub-linear is typically unable to account for assortment effects in a principled fashion, which typically results in a slew of ad-hoc algorithmic tweaks. Here, we seek to begin bridging this gap.

3.1.1. Our Contributions

The present paper seeks to develop sub-linear time algorithms for offer-set optimization while allowing for rich, economically grounded models of customer choice. Specifically, like is typical in the ANN paradigm, we are endowed with a metric space. We are given a universe V of n products, wherein each product $v \in V$ is represented as a fixed point in the metric space (we will describe later on common ways for estimating such an embedding). Similarly, a user U is a *random* point in this space; allowing for U to be random is key to modeling user behavior in a way that is congruent with established models of choice.

Our objective then is to solve, in sub-linear time, a problem of the form

$$\max_{S \subset V, |S| \leq k} \mathsf{E}\left[f(S, U)\right],$$

where the decision space is the subsets of products of cardinality at most k, and the expectation is over U. The principle assumptions we place on the functions $f(\cdot, U)$ are that we require these functions be sub-modular, and further that $f(\{v\}, U)$ be non-increasing in the distance $d(U, v)$.

As we discuss later, this framework is quite flexible: for instance, it immediately captures the problem of picking an offer set to maximize conversion (i.e. the probability of a purchase) where consumer choice is driven by an essentially arbitrary random utility model.

With respect to this model, we make the following contributions:

1. **A Sub-linear Time Algorithm:** Our primary contribution is a sub-linear time algorithm to solve the optimization problem above with uniform performance guarantees. Our algorithm relies on a procedure for constructing, in sub-linear time, a particular sub-linear sized subset of products. This set enjoys the property that the optimal value of our optimization problem restricted to this set is close to the optimal value of the optimization problem over all products. As such, we then simply solve our optimization problem over this restricted set. A greedy algorithm trivially guarantees both a sub-linear run-time and a constant factor approximation.

2. **A New Sampling Scheme:** Our key algorithmic contribution is our approach to constructing the sub-linear set of candidate products, which we dub *locality-sensitive sampling*. Locality-sensitive sampling is a simple idea motivated by the same locality-sensitive hash functions that underly ANN algorithms. By re-interpreting the standard near neighbor problem as one of *sampling* items according to a specific decreasing function of their distances from a query point, we are able to solve the same problem for arbitrary decreasing functions. This generalized sampling problem, along with our sub-linear time solution, may be of independent interest.

3. **Empirical Evaluation:** We present an empirical study on a large-scale corpus of real page-view data from the online advertising platform Outbrain. The dataset contains two billion page views of seven hundred million unique users. Our experiments establish the value of our procedure over existing sub-linear time heuristics, and in particular, that (a) our model of user choice is more accurate in predicting user behavior than the models implicitly assumed by these heuristics, and (b) our algorithm outperforms these heuristics in terms of conversion rate.

The rest of this paper is organized as follows: we review related work in the remainder of this section. Section 2 introduces our problem formally, along with our key modeling assumptions. Section 3 describes the motivation for our algorithm by way of an idealized sampling procedure. We then describe our actual algorithm, which is designed to approximate this idealized procedure,

in Section 4. Experimental results are described in Section 5, and finally conclusions are drawn in Section 6.

3.1.2. Related Work

This work is related to three existing streams of literature, as we describe now.

Recommendation Algorithms: In the area of recommender systems, the problem of learning user preferences from previous interactions has been studied extensively (Jin et al. (2003, 2002), Freund et al. (2003), Schapire and Singer (1998)). For the most part, successful learning algorithms work by embedding both users and items within some metric space such that a user's affinity toward an item is inversely related to their pairwise distance. See Adomavicius and Tuzhilin (2005) for an extensive survey of content-based, collaborative and hybrid recommendation approaches, and Zhang et al. (2019) for a survey of modern approaches based on deep learning.

A recent problem in this stream of literature is how to capture the impact of *diversity* in recommendations. These efforts have mostly focused on quantifying and maximizing diversity in recommendations sets. Kunaver and Požrl (2017) is an extensive survey of the research in this area. A key limitation of the current research here is that the diversity metric is not standardized. Moreover, increasing diversity has often been viewed as sacrificing accuracy of the recommendation set. We will take a more systematic approach to this.

Assortment Optimization: Another stream of literature related to our work is assortment optimization in the field of operations management. Assortment optimization is a principled modeling approach to choosing an optimal assortment to offer to customers. Kök et al. (2008) provides an overview of models found in literature and approaches common in practice. Integral to the assortment optimization problem is the model for user choice. One of the most well studied and commonly used choice model is the Multinomial Logit (MNL) model. The assortment optimization problem with the MNL choice model is tractable, even under various constraints (Talluri and Van Ryzin (2004), Rusmevichientong et al. (2010a), Davis et al. (2013)). Though being attractive due to its tractability, the MNL models suffers from Independence of Irrelevant Alternatives (IIA) property. To overcome the IIA limitation, the Nested Logit (Williams (1977)) and Mixed Multinomial Logit models were proposed. More recently, assortment optimization has been studied under some new choice models like the Markov chain choice model (Désir

et al. (2015)), distance-comparison based choice model (Kleinberg et al. (2017)), distribution over rankings (Farias et al. (2013)) and its variations (Désir et al. (2016)). One limitation of this stream of work is that sub-linear time algorithms effectively do not exist. Even linear time algorithms are rare and restricted to models like the simple multinomial logit that fail to capture user diversity.

Approximate Nearest Neighbors: The third stream of literature that is relevant to our work is the problem of nearest neighbor (NN) search. In this problem, the goal is to pre-process the given data set so that the nearest neighbor to a query can be efficiently calculated. Chávez et al. (2001) give an overview of methods that have been proposed to solve this problem. Some sample works on the NN search problem are Omohundro (1989), Sproull (1991), Bentley (1975), and Yianilos (1993).

We focus on a particular type of approximate nearest neighbor search algorithm called Locality Sensitive Hashing (LSH) (Andoni and Indyk (2008)). Paulevé et al. (2010) describe various hash functions used in LSH algorithms. These have been used in several applications, but in particular find themselves used extensively in recommendation systems. This application has largely focused on obtaining binary representations of users and items which can then be used for doing fast similarity search computations (Karatzoglou et al. (2010), Zhou and Zha (2012), Liu et al. (2014), Das et al. (2007), Liu et al. (2018), Zhang et al. (2014)).

3.2. Model and Assumptions

We begin by introducing the core optimization problem that will be the subject of the rest of this paper. The problem is to select a personalized offer set that maximizes expected reward, subject to a cardinality constraint. Let V denote the universe of items or products we can offer, and $k \in \mathbb{N}$ the maximum cardinality allowed, meaning the set of feasible offer sets is $\{S \subset V : |S| \leq k\}$. The need for personalization is driven by the notion of a user 'type': we assume that each user has a type, which takes values in some set $\mathcal{M}$, and that this type governs the reward we obtain for offering a given offer set. That is, the reward function, which we denote $f(\cdot, \cdot)$, is a map from $2^V \times \mathcal{M}$ to $[0, 1]$, where w.l.o.g. the reward is bounded above by 1. To fix a concrete running example, consider the problem of online content recommendation: the items are webpages, and $f(S, u)$ is the *conversion* probability, i.e. the probability that a user of type u will visit at least one of the webpages in S if they are recommended together – we will expand

on this example later in this section.

To summarize so far, if we were given a user of type $u \in \mathcal{M}$, we would seek to solve the problem $\max_{S \subset V, |S| \leq k} f(S, u)$. We will see later that this problem is 'easy' in many reasonable settings. Instead, one of the two primary challenges we seek to address in this paper is how to deal with *user heterogeneity*, i.e. when the user type is not known exactly ex-ante. We will assume that this uncertainty is modeled as a random variable U over $\mathcal{M}$, whose distribution we know. Our goal then is to solve the following stochastic optimization problem:

$$\text{OPT} \equiv \max_{S \subset V, |S| \leq k} \mathsf{E}\left[f(S, U)\right]. \tag{3.1}$$

For the rest of this paper, we will take U to be uniformly distributed over m types: $u_1, \ldots, u_m \in \mathcal{M}$. There are two motivations for this: first, for m sufficiently large, this assumption is without loss, as replacing the expectation in (3.1) with a sample average approximation results in negligible loss. Lemma B.1.1 in Appendix B.1 shows that $m = \Omega(k \log n)$ is sufficiently large, where $n \equiv |V|$ is the number of items. Second, in practice, what is quite often done to model U is that past observations of a given user are mapped to points in $\mathcal{M}$, and U is taken to be a distribution whose support is over these points; the uniform distribution is one natural choice (we will provide experimental evidence for this in Section 3.5.1).

As was described in the Introduction, we seek to solve (3.1) in online settings in which the number of items n is massive and the optimization must be performed extremely fast, often so fast that even algorithms linear in n are too slow. Thus, the second primary challenge we face is to solve, or approximate, problem (3.1) in *sub-linear* time: $o(n)$. This will require three assumptions, which we will describe in detail in the remainder of this section. Precisely, imposing these assumptions will allow us to guarantee (in expectation) an approximation of OPT using a randomized algorithm whose expected runtime, amortized over multiple users, is $O(n^{1-\epsilon})$ for some strictly positive ϵ.

Our first assumption is that the reward function for any user type be monotone submodular:

Assumption 3.2.1. For every $u \in \mathcal{M}$, the function $f(\cdot, u)$ is monotone submodular.

The set of reward functions satisfying Assumption 3.2.1 is rich enough to include the conversion function for recommendation problems and a subclass of assortment optimization problems against the mixed multinomial logit choice model. Making this assumption is the first step in achieving sub-linearity. In fact, Assumption 3.2.1 already implies that $(1 - 1/e)$OPT

can be guaranteed in *linear* time, as the greedy algorithm is $(1 - 1/e)$-optimal for maximizing monotone submodular functions subject to cardinality constraint (Nemhauser et al. (1978)). Since sums of monotone submodular functions are monotone submodular, the greedy algorithm for (3.1) achieves $(1 - 1/e)\text{OPT}$. Our eventual algorithm will achieve a strictly lower (but still constant) approximation guarantee, but will improve on the greedy algorithm's $O(kmn)$ runtime.

3.2.1. User and Item Embedding

The remaining two assumptions we make will allow us to use the machinery of approximate nearest neighbor algorithms in order to improve from linear to sub-linear time. To discuss these, we will first need to describe the underlying geometry of our problem. Recall that we model user types as elements of a set $\mathcal{M}$, which so far is an arbitrary set. We will assume that $\mathcal{M}$ is in fact a metric space, equipped with a metric denoted by $d(\cdot, \cdot)$. We will also assume that our universe of items is embedded in this same space: $V = \{v_1, \ldots, v_n\} \subset \mathcal{M}$.

Such embeddings are ubiquitous in predictive algorithms for personalization which, by and large, operate by estimating feature (or latent-factor) representations of users and items so that, loosely speaking, a user will have a stronger preference for items whose features 'align' more closely to his or her own features (or equivalently, those items whose features are closer in distance with respect to a carefully calibrated metric). The metric space will often either be Euclidean space or the unit-ball in Euclidean space, but using the Euclidean metric is not a requirement. Instead, what we will need to assume about our space is that there exists an appropriate data structure that returns approximate near neighbors in sub-linear time:

Assumption 3.2.2. For any distance $\gamma > 0$, and constants $c > 1$, $\beta \in [0, 1)$, and $\epsilon \in (0, 1]$, there exists a (randomized) data structure $\text{ANN}[V, \gamma, c, \beta, \epsilon] : \mathcal{M} \to 2^V$, and a corresponding $\alpha \equiv \alpha(c, \beta, \epsilon) < 1$ such that, given any query point $u \in \mathcal{M}$:

1. If
$$\sum_{v \in V} \mathbb{1}(d(v, u) \leq c\gamma) \leq n^\beta,$$
then for each $v \in V$ such that $d(v, u) \leq \gamma$,
$$\mathsf{P}\left(v \in \text{ANN}[V, \gamma, c, \beta, \epsilon](u)\right) \geq 1 - \epsilon.$$

2. The runtime of querying this data structure is $O(n^\alpha)$.

Here, constants suppressed by the big-Oh notation depend only on $\mathcal{M}$ (e.g. dimensionality).

Assumption 3.2.2, while perhaps unusual at first, is stated in the form typically taken in theoretical guarantees for approximate near neighbor algorithms. In words, the data structure assumed here takes any point in $\mathcal{M}$ as input, and outputs every item in V that is within a pre-specified distance γ of the input, each with sufficiently high probability. Most importantly, the runtime of this operation is sub-linear (since α is assumed to be less than 1), assuming that the number of these near neighbors itself is sub-linear (since β is assumed to be less than 1). As a sanity check, if ϵ could be taken to be 0, and c could be taken to be 1, this would correspond to an exact near neighbor algorithm. Having $\epsilon > 0$ reflects the fact that near neighbors are only guaranteed to be returned with high probability. Having $c > 1$ reflects that in the process, elements of V slightly further than γ are generated as candidates and need to be pruned.

The study of approximate near neighbor algorithms has produced data structures satisfying Assumption 3.2.2 for a variety of metric spaces, including Euclidean space. As we will review later on, one way of construct such a structure uses a family of so-called locality-sensitive hash functions with certain 'nice' properties. Our ability to solve (3.1) in sub-linear time will rely on our ability to sample from a certain distribution on $\mathcal{M}$. The algorithm we develop will rely on a carefully constructed ensemble of data structures of the type defined by Assumption 3.2.2

Finally, while Assumptions 3.2.1 and 3.2.2 deal with the reward function f and the underlying metric space $(\mathcal{M}, d)$ separately, there has so far been nothing rigorously tying the two together which would allow us to leverage the metric structure. This is the purpose of our final assumption, which states that the distance between the embeddings of a user and an item directly encodes the corresponding reward for offering that item alone to that user:

Assumption 3.2.3. There exists a non-increasing function $p : \mathbb{R}^+ \to [0, 1]$ such that

$$p(d(v, u)) \geq f(\{v\}, u) \text{ for each } u \in \mathcal{M},\ v \in V.$$

In addition, there exists some $\beta \in [0, 1)$ and $c > 1$ such that

$$\sum_{v \in V} p\left(\frac{d(v, u)}{c}\right) \leq n^{\beta} \text{ for each } u \in \mathcal{M}.$$

The function $p(\cdot)$ captures the inverse relation between distance in $\mathcal{M}$ and reward, and will play a crucial role in our algorithm. In particular, we will treat $p(\cdot)$ as a probability in a

sampling-based approach. The second part of Assumption 3.2.3, which will allow us to make use of the approximate near neighbor in sub-linear time, assumes that for all user types u, the total reward gotten by offering each item individually is sub-linear. This may, for example, reflect the fact that users' appetites for content are not limited by a lack of items, but rather a limit in time, attention, etc. Additionally, this condition is required to be robust in the following sense: there exists some $c > 1$ such that the condition above still holds if each $v \in V$ is replaced by a contracted vector $\tilde{v}$ such that $d(\tilde{v}, u) = d(v, u)/c$.

3.2.2. Examples

We conclude this section by describing two common models which fit the framework we have outlined and satisfy our three assumptions.

Conversion Under Random Utility Choice Models: As described previously, the goal in recommendation problems is typically to induce *conversion*, i.e. selecting at least one of the items in the offer set (e.g. clicking on one of a set of web links, or listening to one of a list of songs). In this setting, $f(S, u)$ is a *conversion function* which, for any set of items $S \subset \mathcal{V}$, is the probability of conversion when customer u is offered set S.

Random utility models are commonly used to describe user choice behavior. One generic way of employing these models in the conversion problem is to assume that $f(S, u)$ takes the form

$$f(S, u) = \mathsf{P}\left(\max_{v \in S}(\mu(d(v, u)) + \epsilon_v) > \epsilon_\emptyset\right), \tag{3.2}$$

where $\mu : \mathbb{R}^+ \to \mathbb{R}^+$ is a non-increasing function, and the ϵ_v's and $\epsilon_\emptyset$ are i.i.d. mean-zero random variables. Here, $\mu(d(v, u)) + \epsilon_v$ is the random utility associated with selecting item v, with $\mu(\cdot)$ translating distance to a mean utility, and ϵ_v capturing idiosyncratic noise. The random utility of selecting no item is $\epsilon_\emptyset$, assuming w.l.o.g. that the mean utility of this option is zero. Users then choose the option (either one of the recommended items, or no item) that maximizes their utility, and (3.2) is the probability that the utility of any recommended item is higher than the utility of selecting nothing.

The formulation in (3.2) satisfies Assumption 1 immediately, and the function $p(\cdot)$ required by Assumption 3 can be constructed from the distribution of the ϵ's. This setup is extremely general and encodes many popular choice models. For example, taking the ϵ's to be Gumbel random variables yields the multinomial logit model, and allowing for random U yields the

mixed multinomial logit.

One commonly used choice of metric space and utility function in the conversion problem is the following: let $\mathcal{M} = \S^{d-1}$, i.e. the unit ball in d-dimensional Euclidean space. As stated earlier, this space satisfies Assumption 2. Now since we are dealing with unit-vectors, we have that $d(v,u)^2 = 2(1 - v^\top u)$, so taking $\mu(x) = 1 - x^2/2$ yields a natural form for the mean utility:

$$\mu(d(v,u)) = v^\top u.$$

Finally, it is worth noting that this formulation is compatible with a number of approaches to constructing metric space representations of users and products, ranging from simple logistic regression, to collaborative filtering, to state of the art approaches such as factorization machines (Rendle (2010)) and field-aware factorization machines (Juan et al. (2016)), the latter having been a key component in the winning entries of three major recent public prediction competitions.

Assortment Optimization Under the Mixed Multinomial Logit Model: In operations management, a classic problem is to select an assortment of products to offer to customers so as to maximize expected *revenue*. Changing the objective from conversion to revenue yields a far more difficult problem, especially given the rich set of choice models and additional operational constraints one could assume. Our model and algorithm will in no way offer a completely general sub-linear time solution, but there do exist meaningful instances to which they can be applied, as we illustrate by example now.

Let r_j be the revenue gained if a customer purchases product v_j. Here, we will just work out the setting where the underlying choice model is the multinomial logit:

$$f(S,u) = \sum_{v_j \in S} r_j \frac{\exp(v_j^\top u)}{w + \sum_{v \in S} \exp(v^\top u)}, \tag{3.3}$$

where $w \geq 0$ is a parameter that controls the likelihood that no product is selected. The objective in (3.3) is not in general monotone or submodular, but there are a variety of conditions which imply both. For example, one such condition shown in Han et al. (2019) is if the minimum and maximum revenues (denoted $r_{\min}$ and $r_{\max}$) are not too far apart:

$$\frac{r_{\min}}{r_{\max}} \geq \max_{S \subset V, |S| \leq k} \sum_{v_j \in S} \frac{\exp(v_j^\top u)}{w + \sum_{v \in S} \exp(v^\top u)}.$$

The expression on the right-hand side is equal to the maximum conversion (as defined previously) probability for a user of type u across all feasible assortments. In particular, the revenues are allowed to vary more when this quantity is small, or equivalently, when w is large. The required upper bound on $f(\{v\}, u)$ can be gotten by treating all revenues as $r_{\max}$.

3.3. Algorithm Overview

Before describing our approach, we could first consider whether some sort of brute force pre-computation would suffice, that is, simply solving (3.1) in advance for a sufficiently comprehensive set of distributions U. If feasible, this would certainly qualify as an amortized sub-linear (constant, in fact) time algorithm. There are at least two reasons why this approach might be infeasible or at best impractical. First, the size of a 'comprehensive' set of distributions U could be massive – even having made our reduction so that U is uniformly distributed over m points of V, this set is of size $O(n^m)$ – in which case it may be practically impossible to compute and/or store it. Second, in almost all settings, the product set is dynamic. For example, the universe of online content is constantly changing. Thus, the data structure needs to be dynamic, ideally capable of fast additions and deletions. The structure we describe in the next section will allow these dynamic updates in sub-linear time; brute force pre-computation would not.

At a high level, our algorithm proceeds in two steps:

(a) Randomly sample a sub-linear sized subset of V, which we will denote by $\tilde{V}$, such that if (3.1) is solved over $\tilde{V}$ instead of V, we are still guaranteed a constant fraction $(1-\epsilon)$ of OPT in expectation.

(b) Approximately solve (3.1) over $\tilde{V}$ using the greedy algorithm.

The crux of our algorithm is the ability to perform step (a) in sub-linear time. Assuming that step (a) could be performed in sub-linear time, the greedy algorithm in step (b) would then also run in sub-linear time, and the algorithm as a whole would be guaranteed $(1-1/e)(1-\epsilon)$ of OPT in expectation.

Ignoring the runtime of step (a) for a moment, we will first describe an idealized random sampling scheme over the items of V that would return a random subset $\tilde{V}$ that is both sub-linear in size and guaranteed (in expectation) to preserve a constant fraction of OPT (less a small additive error) when optimized over. To ease notation, we will fix a distribution U and denote the objective function of (3.1) by $g(S) = \mathsf{E}[f(S, U)]$.

Suppose that we could randomly sample $\tilde{V}$ such that

$$\mathsf{P}(v \in \tilde{V}) = g(\{v\}) \text{ for each } v \in V. \tag{3.4}$$

That is, the likelihood of any item being included in $\tilde{V}$ is equal to the reward that the item would yield when offered alone (recall that this reward is assumed w.l.o.g. to lie in $[0, 1]$). The following Lemma shows that making sufficiently many independent draws from such a sampling distribution, and taking the union of these draws, would result in a subset of V that is guaranteed a constant fraction of OPT if subsequently optimized over:

Lemma 3.3.1. For $c \in (0, 1]$, let $\tilde{V}$ be a random variable taking values in 2^V such that

$$\mathsf{P}(v \in \tilde{V}) \geq cg(\{v\}) \quad \text{for each } v \in V,$$

and for $s \in \mathbb{N}$, let $\tilde{V}_s$ denote the union of s sets drawn i.i.d. from this distribution.

Let $S^*(\tilde{V}_s)$ be an optimal solution to:

$$\max_{S \subset \tilde{V}_s, |S| \leq k} g(S).$$

Then for any $\epsilon_1, \epsilon_2 \in (0, 1]$, if

$$s \geq \frac{k}{c\epsilon_2} \log \frac{k}{\epsilon_1},$$

then we have

$$\mathsf{E}[g(S^*(\tilde{V}_s))] \geq (1 - \epsilon_1)\text{OPT} - \epsilon_2.$$

Proof of Lemma 3.3.1. Fix any $\delta \in [0, 1]$ (we will tune this quantity in the end). For any $v \in V$ such that $g(\{v\}) \geq \delta$, we have

$$\begin{aligned} \mathsf{P}(v \notin \tilde{V}_s) &= \mathsf{P}(v \notin \tilde{V})^s \\ &\leq (1 - cg(\{v\}))^s \\ &\leq (1 - c\delta)^s \\ &\leq e^{-sc\delta}, \end{aligned} \tag{3.5}$$

where the first equality follows from the definition of $\tilde{V}_s$, and the first two inequalities are by assumption.

Now we fix any optimal solution S^* to the full problem (3.1), and divide it into two disjoint sets:

$$S_1 = \{v \in S^* : g(\{v\}) \geq \delta\} \text{ and } S_2 = \{v \in S^* : g(\{v\}) < \delta\}.$$

Then we have

$$\begin{aligned} g(S^*(\tilde{V}_s)) &\geq g(S_1 \cap \tilde{V}_s) \\ &\geq \mathsf{P}(v \in \tilde{V}_s \ \ \forall\, v \in S_1) g(S_1) \\ &\geq \left(1 - \sum_{v \in S_1} \mathsf{P}(v \notin \tilde{V}_s)\right) g(S_1) \\ &\geq \left(1 - ke^{-sc\delta}\right) g(S_1) \\ &\geq \left(1 - ke^{-sc\delta}\right) (g(S^*) - g(S_2)) \\ &= \text{OPT} - ke^{-sc\delta}\text{OPT} - \left(1 - ke^{-sc\delta}\right) g(S_2), \end{aligned} \tag{3.6}$$

where the first line is due to the optimality of $S^*(\tilde{V}_s)$ among all solutions contained in $\tilde{V}_s$, the third line is a union bound, the fourth line is due to (3.5), and the fifth line is due to submodularity.

To conclude, it will suffice to upper bound the second and third terms in (3.6) by ϵ_1OPT and ϵ_2, respectively. To do this, we choose $\delta = \epsilon_2/k$. For the second term, applying this choice of δ, along with our condition on s, yields the following bound:

$$ke^{-sc\delta}\text{OPT} \leq ke^{-\log(k/\epsilon_1)}\text{OPT} = \epsilon_1\text{OPT}.$$

For the third term, by submodularity and the definition of S_2,

$$\left(1 - ke^{-sc\delta}\right) g(S_2) \leq g(S_2) \leq \sum_{v \in S_2} g(v) \leq k\delta = \epsilon_2.$$

■

Lemma 3.3.1 shows that to approximate OPT to arbitrary precision in expectation, it suffices to sample $s = O(k \log k)$ times from a distribution approximately satisfying (3.4). In fact, the Lemma states that the sampling probabilities do not need to match (3.4), but that they just need to be lower bounded by some constant fraction c of (3.4). In the algorithm we outline later, we will arbitrarily take this fraction to be $c = 1/2$.

Having guaranteed that a constant fraction of OPT is preserved, the other required condition on (3.4) is that the resulting subset be sub-linear in size, as the greedy algorithm that follows is linear in the size of this set. Fortunately, the expected size of $\tilde{V}$ sampled according to (3.4) is guaranteed to be sub-linear:

$$\mathsf{E}[|\tilde{V}|] = \sum_{v \in V} g(\{v\}) = \mathsf{E}\left[\sum_{v \in V} f(\{v\}, U)\right] \leq \mathsf{E}\left[\sum_{v \in V} p(d(v, U))\right] \leq n^{\beta}, \tag{3.7}$$

where both inequalities relied on Assumption 3.2.3.

3.4. Our Approach in Detail: Locality-Sensitive Sampling

To recap, the main conclusion drawn from the previous section is that the ability to sample according to (3.4) in sub-linear time is sufficient for our goal of constructing an algorithm that is itself sub-linear and that achieves a constant approximation of OPT. In this section, we will first describe our solution to this sampling problem, which makes use of the abstract ANN data structures assumed to exist in Assumption 3.2.2. We will then, as a slight detour (which may be skipped), describe what a concrete version of this approach looks like using actual locality-sensitive hash functions. Finally, we close the loop and provide our theoretical guarantee in the form of Theorem 3.4.4.

3.4.1. Approximating the Ideal Sampling Distribution via Locality-Sensitive Sampling

Now with the goal of executing the ideal sampling distribution (3.4), the brute-force method to sample exactly from this distribution would be to generate n independent Bernoulli variables, whose means are $g(\{v\})$ for each $v \in V$. By Lemma 3.3.1, repeating this procedure $s = \Omega(k \log k)$ times yields a pruned set of items that preserves a good approximation of OPT in expectation. However, generating the mn Bernoulli variables is clearly a linear time procedure. Fortunately, it is possible in sub-linear time to *approximate* (3.4), i.e. the probabilities in (3.4) are not matched exactly, but rather just up to a constant:

$$\mathsf{P}(v \in \tilde{V}) \sim g(\{v\}) \text{ for each } v \in V.$$

The key observation that makes this possible is that while the probabilities of the individual events $\{v_j \in \tilde{V}\}$ need to be strictly controlled, these events are allowed to be arbitrarily correlated. This is precisely what allows us to leverage the underlying metric space, along with the approximate near neighbor data structures assumed by Assumption 3.2.2, to approximately perform (3.4).

To see why this is possible, first note that allowing arbitrary correlations implies that to sample each v with probability $g(\{v\}) = \mathsf{E}[f(\{v\}, U)]$, it suffices to first draw u according to U, and then sample each v with probability $f(\{v\}, u)$. Next recall that by Assumption 3, there exists a function $p(\cdot)$ such that $f(\{v\}, u) \leq p(d(v, u))$. This allows us to use near neighbor queries to do the sampling As a simple example, if $p(x) = \mathbb{1}(x \leq \gamma)$ for some γ, then sampling with probability $f(\{v\}, u)$ is equivalent to returning all $v \in V$ within distance γ of u, and thus a single approximate near neighbor data structure suffices.

More generally, our algorithm utilizes $R = \lfloor \log_2 n^{1-\beta} \rfloor$ of these structures to approximate any non-increasing function $p(\cdot)$. For each $r \in [R]$, let

$$\rho_r = \begin{cases} 1/(2^r - 1), & r \in [R-1] \\ 1/2^{r-1}, & r = R \end{cases} \qquad \text{and} \quad \gamma_r = \sup\{x : p(x) \geq 1/2^r). \tag{3.8}$$

See Figure 3.1 for a visual depiction of these parameters and our strategy, which is to approximate $p(\cdot)$ by a step function, each step represented by a single near neighbor data structure.

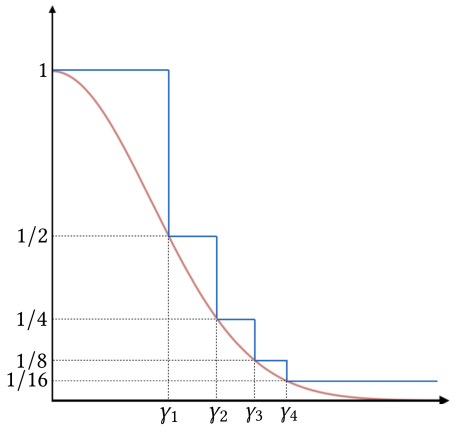


Figure 3.1: Visual depiction of the approximate sampling scheme. The red curve contains the ideal sampling probabilities $p(\cdot)$, and the blue curve shows how we attempt to approximate it using a step function. In this example, $R = 4$.

Our overall scheme then, defined formally below, is to create a set of approximate near neighbor structures, and sample from $p(d(v, u))$ by querying each structure and returning their union. The various parameters for these structures are given by the ρ_r's and γ_r's, along with

our choice of $\epsilon = 1/2$ (chosen arbitrarily to save on notation).

Definition 3.4.1 (Locality-Sensitive Sampling)**.** Let V be a finite subset of a metric space $\mathcal{M}$ satisfying Assumption 3.2.2, For any non-increasing function $p : \mathbb{R}^+ \to [0, 1]$, and any constant $c > 1$, the *Locality-Sensitive Sampling* data structure is a (randomized) map $\text{LSS}[V, p, c, \beta] : \mathcal{M} \to 2^V$:

$$\text{LSS}[V, p, c, \beta](u) = \rho_0 V \bigcup \left(\bigcup_{r=1}^{R} \text{ANN}[\rho_r V, \gamma_r, c, \beta, 1/2](u) \right) \quad \text{for all } u \in \mathcal{M},$$

where the ρ_r and γ_r are defined as in (3.8), $\rho_0 = \frac{1}{2} n^{\beta - 1}$, and each $\rho_r V$ denotes a random subset of V gotten by including each element of V independently with probability ρ_r.

The locality-sensitive sampling data structure achieves the approximate sampling distribution we seek in sub-linear time. This is stated formally in the following Lemma, whose proof appears in Appendix B.2.1.

Lemma 3.4.2. For each $u \in \mathcal{M}$ and $v \in V$,

$$\mathsf{P}(v \in \text{LSS}[V, p, c, \beta](u)) \geq p(d(v, u))/2.$$

Moreover, each query $\text{LSS}[V, p, c, \beta](u)$ has runtime

$$O\left(n^{\alpha} \log n\right),$$

where $\alpha = \alpha(c, \beta, 1/2)$.

3.4.2. Aside: LSS Using Locality-Sensitive Hash Functions

So far, we have assumed the existence of ANN data structures satisfying Assumption 3.2.2, without describing how any of these work. In this subsection, we describe one existing approach based on locality-sensitive hash (LSH) functions (originally described in the seminal work of Indyk and Motwani (1998)), and show how an LSS structure can be constructed from scratch from these functions. This subsection can be safely skipped without loss of continuity.

The key component of LSH algorithms are *LSH families*: let $\mathcal{H}$ be a family of functions defined on $\mathcal{M}$ such that when h is chosen uniformly at random from $\mathcal{H}$, we have $\mathsf{P}(h(u_1) = h(u_2)) = q(d(u_1, u_2))$. Here, $q : [0, \infty) \to [0, 1]$ is some non-increasing function such that $q(0) = 1$

and $q(x) > 0$ if $p(x) > 0$. We will show that sampling from V can be approximated in sublinear time using an LSH family $\mathcal{H}$ and our Locality-Sensitive Sampling procedure:

Proposition 3.4.3. Let $\mathcal{H}$ and q be defined as in the preceding text, and suppose that

$$\log_{q(cx)} q(x) \leq \delta \quad \text{for all } x \text{ and some } \delta < 1.^{1}$$

Then there exists a locality-sensitive sampling data structure built from these hash functions such that Lemma 3.4.2 holds with

$$\alpha = \beta + \delta(1 - \beta).$$

The proof can be found in Appendix B.2.2. Proposition 3.4.3 is only useful assuming the existence of a family of functions $\mathcal{H}$ satisfying the condition in the statement of the Proposition. Does such a family in fact exist? The search and analysis of appropriate families of functions for various metric spaces has been an active area of research. For our own setting, where $\mathcal{M} = \mathbb{S}^{d-1}$ and the metric is induced by the ℓ_2 norm, there are recent results (Terasawa and Tanaka (2007), Andoni and Razenshteyn (2015), Andoni et al. (2015)) for the *cross-polytope* hash family that essentially amounts to randomly rotating a set of pre-defined points on the sphere and hashing each vector to its nearest point. Even simpler is the *hyperplane* LSH family where each function corresponds to a single vector, and the function assigns to any vector the sign of its inner product with the defining vector. Charikar (2002) show that δ can be taken to be $1/c$ using this family.

To describe the locality-sensitive sampling procedure, we begin by defining the vanilla LSH data structure, which we parameterize by $\rho \in [0, 1]$ and integers $a, b > 0$. To construct the data structure, first a random subset $\rho V \subset V$ is taken by including each element of V independently with probability ρ. Then a total of b hash tables are constructed, with each table storing all of the items in ρV. The hash function for each table $j = 1, \ldots, b$ is vector-valued, constructed by drawing functions $h_1^j, \ldots, h_a^j$ independently and uniformly at random from $\mathcal{H}$. This entire construction is done during the preprocessing phase. Then given a query point $u \in \mathcal{M}$, we hash u in each table and return all collisions:

$$\mathrm{LSH}_{\rho,a,b}(u) = \left\{ v \in \rho V : (h_1^j(v), \ldots, h_a^j(v)) = (h_1^j(u), \ldots, h_a^j(u)) \text{ for some } j \in [b] \right\}.$$

[1] We follow the convention that $\log_0 x = 0$ for any $x \in [0, 1]$.

Thus, $\text{LSH}_{\rho,a,b}(u)$ is a random subset of V, where the randomness is with respect to the sampling when creating ρV and selecting the hash functions.

Our locality-sensitive sampling algorithm then utilizes $R = \lceil 1 + \log_2 n \rceil$ LSH structures. For each $r \in [R]$, let ρ_r and γ_r be defined as in (3.8). Moreover, let

$$a_r = \left\lceil \log_{q(c\gamma_r)} 2^r n^{\beta-1} \right\rceil \quad \text{and} \quad b_r = \left\lceil \log(2) 2^{-r\delta} n^{\delta(1-\beta)} (1/q(\gamma_r)) \right\rceil .$$

Then given a query u, for each $r \in [R]$, we calculate $\text{LSH}_{\rho_r,a_r,b_r}(u)$ and return their union:

$$\rho_0 V \bigcup \left(\bigcup_{r=1}^{R} \text{LSH}_{\rho_r,a_r,b_r}(u) \right) .$$

To demonstrate this procedure concretely, Figure 3.2 shows the results of an actual implementation of locality-sensitive sampling on a synthetic dataset (experiments using real data will be described in the following section). This synthetic data consisted of a single query point u, and a set V of 50,000 vectors, all lying on the unit Euclidean ball in dimension 50. The vectors in V were randomly generated in such a way that their distance to u is approximately uniform over $[0, 2]$.

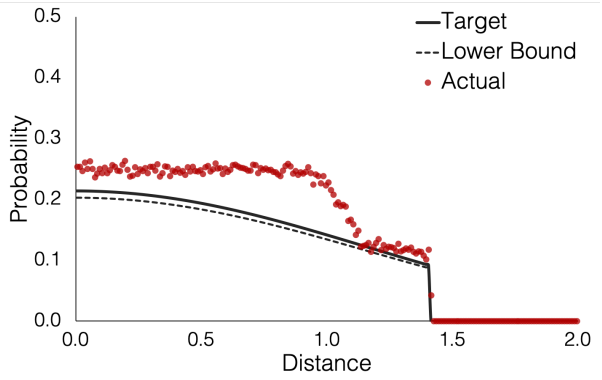


Figure 3.2: Example of sampling of products using Locality-Sensitive Sampling. The sampling probability achieved by the locality sensitive sampling procedure are shown, along with the target sampling distribution and lower bound.

The target sampling probability function $p(\cdot)$, illustrated by the solid black line, corresponds to the conversion rate for a truncated version of the multinomial logit model.[2] This particular locality-sensitive sampling scheme aims to approximate $p(\cdot)$ by sampling each item at a distance x from u with probability at least $0.95p(x)$, as represented by the dotted line. The hash functions

[2]The exact choice was

$$p(x) = \begin{cases} 1 - \frac{10}{10+\exp(1-x^2/2)} & 0 \le x < \theta \\ 0 & x \ge \theta \end{cases}$$

used were generated from the aforementioned Hyperplane LSH family.[3]

To estimate the actual sampling probabilities achieved, we repeated the locality-sensitive sampling procedure 20 times on the dataset (the hash functions were re-chosen randomly in each of these replications leading to different LSH structures), and then measured the fraction of instances for which each item was sampled. Each red point in Figure 3.2 represents the average sampling probability for a 'bin' of about 250 items with nearly equal distance to u. We observe that the locality-sensitive sampling scheme effectively samples as per the desired distribution.

3.4.3. Putting It All Together

Through locality-sensitive sampling, we now have a method of approximating our ideal sampling scheme (3.4). Lemma 3.3.1 requires that $\tilde{V}$ be constructed from $s = \Omega(k \log k)$ *independent* samples from this distribution, so we require s instances of this overall structure (each LSS structure itself a combination of ANN structures).

Our final step then is to solve

$$\max_{S \subset \tilde{V}, |S| \leq k} g(S)$$

using the greedy algorithm, where recall that $g(S) = \mathsf{E}[f(S, U)]$. Specifically, this problem is one of maximizing a monotone submodular set function under cardinality constraint, and as such, is known to admit a $1 - e^{-1}$ approximation via a greedy algorithm (Nemhauser et al. (1978)). Note that the $1 - e^{-1}$ guarantee is the best-known guarantee among polynomial-time algorithms; indeed, even the conversion problem under the no-noise case ($\epsilon = 0$) falls into a class of geometric set cover problems known to be APX-hard (Mustafa et al. (2014)).

The greedy algorithm constructs a solution sequentially as follows: at step ℓ, having already constructed set $S_{\ell-1}$, we choose S_ℓ to be

$$S_\ell = S_{\ell-1} \cup \arg\max_{v \in \tilde{V}} g(S_{\ell-1} \cup \{v\}),$$

where ties are broken arbitrarily. Initiating S_0 to be the empty set, the algorithm completes in k steps. Each step of this greedy algorithm requires calculating $g(S_{\ell-1} \cup \{v\})$ for each v in $\tilde{V}$, with each evaluation taking $O(m)$ time. Therefore, the entire greedy procedure runs in $O(km|\tilde{V}|)$ time. To summarize, the last two sections have shown that our algorithm successfully achieves an approximation in sub-linear time.

[3]The FALCONN (Andoni et al. (2015)) software package was used to build the LSH structures

Theorem 3.4.4. For any $\epsilon_1, \epsilon_2 \in (0, 1]$, there exists a data structure and algorithm that achieves

$$(1 - e^{-1})[(1 - \epsilon_1)\text{OPT} - \epsilon_2]$$

in expectation and has amortized runtime

$$O\left((n^{\alpha} \log n + kmn^{\beta})\frac{k}{\epsilon_2} \log \frac{k}{\epsilon_1}\right),$$

where $\alpha = \alpha(c, \beta, 1/2)$.

3.5. Experiments on Real Data

We performed two sets of experiments which demonstrate that in real applications:

1. **Modeling:** our approach to modeling user behavior, particularly with respect to user diversity (i.e. a stochastic user U), is more accurate than heuristic sub-linear approaches which do not incorporate diversity.

2. **Optimization:** our algorithm outperforms existing sub-linear heuristics in terms of optimizing reward under our model.

These experiments were run using real data from Outbrain, an online advertising platform that provides content recommendations on the websites of numerous publishers (e.g. Figure 3.3). Outbrain serves over 250 billion personalized content recommendations every month and reaches over 565 million unique visitors. Their promoted articles appear on more than 35,000 websites, reaching over 87% of internet users in the U.S. (Outbrain (2017)).

Our dataset contains a sample of pages viewed and clicked on by users on multiple publisher sites in the United States over a two-week period. Specifically, the data contains about two billion page views for 700 million unique users across 560 websites (amounting to around 100GB of data). Before both experiments, we performed an initial pruning to remove pages which had been viewed fewer than 750 times and users who had viewed fewer than 30 pages. There were approximately $174,000$ pages and $640,000$ users remaining.

We then partitioned the users into two groups, one test group (consisting of 10,000 users) to be used for the actual experiments, and one training group (remaining users) to be used for estimating an accurate metric embedding for the pages. The metric embedding was estimated

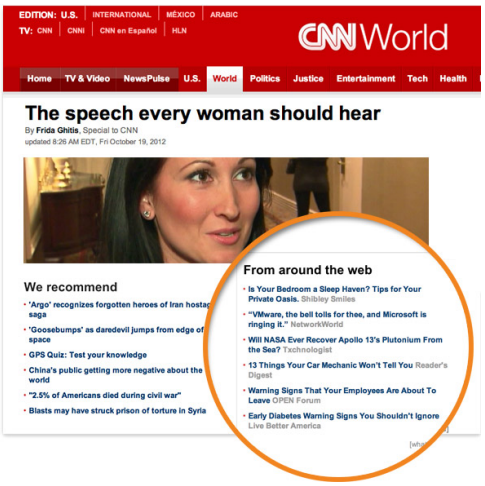


Figure 3.3: Example of content recommendation by Outbrain.

using a model called *word2vec* (Mikolov et al. (2013a,b)): the pages are treated as 'words', and each user's list of viewed pages (in chronological order) is treated as a 'sentence', and the model is trained to predict probabilities of pages appearing near each other (in sentences).[4] The result is a representation of each page in Euclidean space (which we took to be 50-dimensional), and we normalized each vector to lie on the unit sphere. This embedding was used in both experiments.

Finally, we chose to model user choice as a truncated multinomial logit, meaning a fixed user type u, if recommended a set of pages S, has conversion probability

$$f(S,u) = \frac{\sum_{v \in S,\ v^T u > 0} \exp(v^\top u/\sigma)}{w + \sum_{v \in S,\ v^T u > 0} \exp(v^\top u/\sigma)}.$$

Here, σ captures the variability of the ϵ's under the random utility model (3.2) (specifically, it is the scale parameter of the mean-zero Gumbel distribution), and w is a function of the no-choice utility. In both of our experiments, we varied σ from 0.01 to 1, and tuned w for realistic conversion probabilities.

3.5.1. Modeling Diversity in User Behavior

In the coming second set of experiments, we will consider the recommendation problem assuming that each user U is modeled as the uniform distribution over the first 10 pages he or she has viewed in the data. Before considering that optimization problem though, it is worth asking whether this model for U is reasonable. In particular, is it more accurate than models which fix

[4] *word2vec* relies on a two layer neural network. After the training, the weight matrix of the hidden layer of the neural network gives the representation of the words in Euclidean space. Its application in this setting is referred to as *prod2vec* (Grbovic et al. (2015)). We used the implementation of *word2vec* in Python Machine Learning Library (MLlib), built on Apache Spark.

U to be a single point? (The sub-linear heuristics we will soon compare against can be viewed as implicitly assuming such single point models).

To evaluate the accuracy of any model that is given a user's first 10 pages viewed, we measured how predictive it was of the user's behavior after these first 10 pages viewed. We compared our mixture model to two single-point models: *Mean*, which represents U as the average of the first 10 pages viewed, and *Last*, which represents U as the 10th page viewed. Now, for each user, our data contains the pages viewed (the 'positive' samples in a classification task), but unfortunately does not specify when pages were offered to the user and not viewed (the 'negative' samples).

As a reasonable proxy for a dataset with positive and negative samples, we randomly selected a set of pages assumed to have been offered to each user, in a manner that takes into account the 'popularity' of pages. Specifically, for each page j, let T_j denote the number of views of the web page by the training users. Then, a set of exactly 100 pages was randomly sampled such that the likelihood of each page j being included in the set was proportional to T_j^α. The sampling exponent α controls the extent to which higher likelihoods are given to commonly viewed pages. We varied α from 0.2 to 1.0 in our experiments.

			AUC			Average Precision		
σ	w	α	Mixed	Mean	Last	Mixed	Mean	Last
0.01	2.75	0.2	0.89	0.81	0.74	0.17	0.09	0.08
		0.5	0.89	0.81	0.73	0.20	0.12	0.09
		0.7	0.89	0.81	0.73	0.25	0.14	0.11
		1.0	0.89	0.81	0.73	0.33	0.20	0.15
0.1	4	0.2	0.95	0.95	0.90	0.09	0.09	0.07
		0.5	0.95	0.94	0.90	0.11	0.11	0.09
		0.7	0.96	0.96	0.91	0.15	0.15	0.12
		1.0	0.96	0.95	0.91	0.20	0.20	0.16
1	500	0.2	0.94	0.95	0.90	0.08	0.09	0.07
		0.5	0.94	0.95	0.90	0.10	0.11	0.09
		0.7	0.94	0.95	0.90	0.13	0.14	0.12
		1.0	0.95	0.96	0.92	0.18	0.21	0.16

Table 3.1: Accuracy of our model of user behavior (Mixed) compared against two single-point benchmarks (Mean and Last). The area under the ROC curve (AUC) and average precision are reported for these models, for a variety choices of the multinomial logit tuning parameters (σ, w) and sampling exponent (α). Results are aggregated over the entire set of test users, replicated 20 times each.

Given this set of offered pages for each user, the task for each model was to predict which pages

the user had actually viewed. For every page, the models made this prediction by calculating the conversion probability of an assortment containing just that page. The results are reported in Table 3.1. Since this is effectively a classification task, the metrics reported are the area under ROC curve (AUC), and the average precision.[5] Table 3.1 shows that for small to medium-sized values of σ, our mixed model more accurately predicts user behavior than the two single-point models, and for large σ, the accuracy of the mean model is comparable. These results are robust over different choices of α. Finally, the actual AUCs of our mixed model run as high as 0.96, which demonstrates that our model is quite accurate in absolute terms.

3.5.2. Optimization

Finally, to test our proposed optimization algorithm, we again modeled users U as being uniformly distributed over their first 10 viewed pages, this time treating these models as ground truth. Over these choice models, we considered the problem of recommending a set of 10 pages to maximize conversion. We compared our own algorithm (LSS) against two benchmarks that are common practice in reality: *Mean*, which returns the (approximate) nearest neighbors of the mean of the user's first 10 viewed pages, and *Last*, which returns the (approximate) nearest neighbors of the user's 10th viewed page. Both of these benchmarks require a single approximate near neighbor query and are thus sub-linear.

		Avg. Conversion			Win Percentage		
σ	w	LSS	Mean	Last	LSS	Mean	Last
0.01	2.75	0.061	0.041	0.037	0.70	0.18	0.12
	2.78	0.024	0.016	0.015	0.67	0.19	0.15
0.1	4.0	0.064	0.060	0.049	0.52	0.40	0.08
	4.5	0.021	0.020	0.016	0.52	0.38	0.09
1	500	0.042	0.042	0.038	0.25	0.74	0.01
	1000	0.021	0.021	0.019	0.24	0.74	0.02

Table 3.2: Comparison of our algorithm (LSS) to two common practice benchmarks (Mean, Last) on a recommendation problem for mixed multinomial logit users. Each algorithm's conversion rate is reported, averaged over all test users. For each algorithm, the percentage of test users for which that algorithm achieved the highest conversion is also reported.

The results are summarized in Table 3.2. As in the previous set of experiments, we varied σ between 0.01 to 1, and tuned w so that the resulting conversion rates were reasonable. Table 3.2 shows that, for small to medium-sized values of σ, our algorithm outperforms both benchmarks

[5] Both are common metrics for classification, lying in $[0, 1]$, with higher values signifying greater accuracy.

in terms of both the average conversion rate achieved across all test users, and the proportion of test users for which each algorithm was the best. For large-sized σ, the mean benchmark is comparable.

3.6. Conclusion

We proposed a principled approach to offer set optimization that includes (a) a flexible model for user choice that incorporates the underlying structure of commonly estimated item and user metric embeddings, and (b) an algorithm for optimizing offer sets that achieves both a sub-linear runtime and a uniform performance guarantee. Along the way, we developed a sub-linear time algorithm for a certain class of sampling problems that generalizes the classic approximate near neighbor problem and may be of independent interest. Experiments on a real, large-scale dataset from the online advertising platform Outbrain demonstrated the practicality of our modeling, and superiority of our algorithm against common practice benchmarks.

Chapter 4

Multi-Purchase Behavior: Modeling and Optimization

4.1. Introduction

In both online and offline shopping, consumers typically purchase multiple products. Multiple purchases can happen due to complementarity, neighborhood affects (i.e., due to certain items shown or placed next to each other), or due to overlap in the purchase frequencies of various products (sometimes across unrelated categories) (Manchanda et al. 1999). All these effects may not be active simultaneously, and difficult to measure directly. Further, given an expressive choice model that captures this behavior, it is a priori unclear how to optimize for the product recommendations that the platform should ideally show to consumers as these tend to be hard combinatorial optimization problems. The ability to capture a rich enough choice behavior of each consumer as well as to display real-time optimized recommendations to them can have a tremendous impact on the user experience and hence the bottom-line of online shopping platforms. Given that online shopping is one of the most popular activities on the Internet world wide, with sales projected to exceed 6.5+ trillion US dollars by 2022 (Clement 2019), such a personalization effort can yield significant dividends.

Addressing the above challenges, we consider a parsimonious family of multi-purchase choice models (i.e. models capturing purchase of more than one products in an interaction), called the *bundle multivariate logit models* (BundleMVL-K), which is inspired by multiple variations proposed in the marketing literature (Cox 1972, Russell and Petersen 2000, Hruschka et al. 1999, Singh et al. 2005). It models the conditional probability of purchasing a product given the

purchase/no-purchase of all other products and parsimoniously extends the popular multinomial choice model (MNL) to the setting where the customer purchases one or more items (represented by the suffix K $\in \mathbb{Z}_+$ in the acronym) when they are shown a set of recommendations. While modeling multiple purchases has been addressed in the marketing literature, with the end goal of improving customer understanding, the application of BundleMVL-K to recommendation sets of products (not just categories) and subsequent optimization based on it with a focus on its use in web scale settings is new to this work. We contrast BundleMVL-K to existing single-choice and multi-choice models (Benson et al. 2018), deliberate on the optimal choice of K based on practical considerations, and validate the model performance (in terms of likelihood) on eight real datasets. For instance, BundleMVL-K's parameters allow for interpreting substitution behavior and complementary relationships between products. Model fit (assessed using out of sample log likelihood values) when compared against state of the art multi-choice models shows that the BundleMVL-2 (i.e., K= 2) is a strong candidate in capturing rich multi-choice behavior of customers. In particular, we show that the benefit of taking multiple purchases into account in terms of revenue and sales is at least $6-8\%$ in relative terms when compared to the MNL (this is for 1500 products, for other revenue improvement comparisons see Section 4.5). In addition, across 8 real world datasets, the test log-likelihood fits of our multi-purchase model are on average 17% better. Our approach addresses the key future direction discussed in the practice paper by Feldman et al. (2019), who establish strong evidence of the utility of single choice models in a product display setting at the firm Alibaba.

This work is one of the first attempts focused on optimizing revenue for multi-purchase choice models, allowing us to make a link between the model and the revenue gains that it allows. Most prior work, especially in the marketing literature, has not be able to establish this, understandably due to complexity of the revenue optimization problems involved. In particular, assuming that each customer has a distinct BundleMVL-K model associated with them, we develop an iterative binary search based optimization scheme for computing approximately revenue optimal recommendations, while balancing its computational time and solution quality. We complement this by deriving several structural results about the optimal solution that help efficiently explore the search space, as well as by establishing the hardness of the problem (showing that it is indeed NP-complete when $K = 2$, compared to linear time when $K = 1$). The structural results allow our algorithms to determine if a product is part of the optimal solution or not with certainty in constant time, which allows us to work with smaller problem

instances computationally. The binary search based algorithm solves a quadratic unconstrained binary optimization problem in each comparison step (we show how to solve this practically using state of the art heuristic approximation techniques as well). Our approach is compared against a mixed integer programming benchmark, a natural greedy approach that extends the ADXOPT algorithm developed for the MNL single-choice model (Jagabathula 2014) and the revenue-ordered heuristic among others. We also shed light on the properties of the solutions obtained using these benchmarks and when they perform optimally. Our solution to the revenue optimal recommendation problem is one of the first that is also practical and scalable for this class of models. The operational value of our algorithms is that they allow practitioners to capture additional revenue by being able to compute near optimal highly relevant recommendations at scale, hence minimizing the impact on customer disengagement due to computational delays that is typically observed in online platforms (Palmer 2016). Along the way, we also make progress on the theoretical and computational tractability of recommendation set optimization under another natural multi-choice model proposed in (Benson et al. 2018), and illustrate how the BundleMVL-K model and recommendations based on it provide superior performance. Our empirical results on multiple real world datasets strongly support the possibility of real-time personalization that captures rich multi-purchase choice behavior. Table 4.1 outlines the list of our results and algorithmic techniques.

Section 4.2 describes the BundleMVL-K family of choice models. In Section 4.3, we state the revenue optimization problem and derive structural properties of the optimal solution, which we use in developing an iterative binary search based approach (Section 4.4). Extensive numerical experiments are detailed in Section 4.5. A brief discussion in Section 4.6 is followed by a conclusion in Section 4.7. All proofs are delegated to the appendix. Key prior work related to the problem and our solution approach are discussed next.

Results	Section
Model: BundleMVL-2	
- Estimation (Lem. 4.2.1, Alg. 8)	4.2 & appendix
- Hardness (Thm. 4.3.1), and structural properties (Lem. 4.3.3-4.3.5)	4.3
- BinarySearchAO (Alg. 3)	4.4.1
- BinarySearchAO(Efficient)(Alg. 4)	4.4.1
- NoisyBinarySearchAO(Efficient)	4.4.1
- Integer program (Alg. 9)	appendix
- AdxoptL(Alg. 10, Lem. 4.4.1)	appendix
- Revenue-ordered (Thm. 4.4.3 & 4.4.5)	4.4.3
Model: MMC	
- Hardness (Thm. 4.3.2)	4.3.1

Table 4.1: Summary of results.

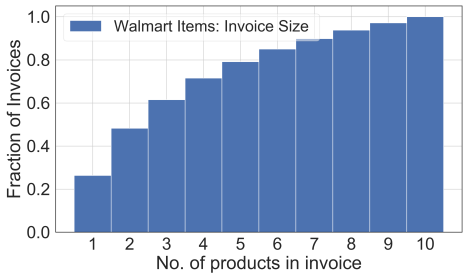


Figure 4.1: CDF of the number of products purchased per transaction in the Walmart dataset.

4.1.1. Relevant Literature

Though purchasing multiple products by customers is extremely common in both online platforms and brick and mortar stores, the majority of research in choice modeling has focused on single product purchases. Train (2009) gives a good overview of the commonly used choice models, which include the MNL model (Plackett 1975), the nested logit model and others. More recently, a variety of other choice models such as the Markov chain choice model, the distribution over ranking modelhave been proposed and studied. Most work here concerns with both the estimation of the model from data, as well as the design of algorithms for revenue maximizing choice sets (assortments) and other related objectives. Some works have pursued robust algorithms, such as the Adxopt (Jagabathula 2014), designed to be choice model agnostic, while others have tried to capture various business-driven constraints, such as a constraint on the number of products that can be recommended, precedence constraints among products etc. (Rusmevichientong et al. 2010a, Sinha and Tulabandhula 2020). We refer the readers to Kök et al. (2008) for an overview of these optimization methods. In many datasets, some of which are explored here, only a minority percentage of transactions reflect single purchases, supporting the need for multi-purchase modeling that better reflect reality.

In the marketing literature, multi-choice models have appeared in the context of modeling purchase of products in multiple categories (Seetharaman et al. 2005) as well as in bundle choice modeling. Two types of choice models have been predominant here - the multivariate probit (MVP) and variants of the multivariate logit (MVL). Both of these models are based on the *random utility theory*. The earliest variation of MVL was proposed in Cox (1972) and

has been used and improved upon in various subsequent works such as Russell and Petersen (2000) and Singh et al. (2005). In bundle choice modeling (McCardle et al. 2007), consumer choice is modeled at the level of product bundles instead of the category of products. Both the MVL variants (Kopalle et al. 1999) and the MVP model have also been considered for this task. Recently, neural network based multi-choice models have also been proposed (Yang and Sudharshan 2019) to capture multi-purchase behavior. While they are able to fit observed data much better than the parametric models considered here, the resulting recommendation set optimization problems become quickly intractable due to lack of structure. Even with parametric models (such as ours), the optimization problems tend to be NP-hard, and in this paper, we devote significant effort to tame this complexity to make recommendation set computations scalable. A distinct problem that parallels our work is that of bundle pricing and optimization, where instead of computing recommendation sets that take choice behavior into account, one models how to price groups of products to maximize sales and revenue, see Ettl et al. (2020) and references therein.

A key prior work is by Benson et al. (2018) who propose a MVL variant, which we refer to as the Mixture Multi-Choice (MMC) model. This model assumes that the mean utility of any subset of products is the sum of the utilities of each product in the subset and an optional correction term. By limiting the number of sets which receive a correction, this model can have a sparse parameterization. A crucial drawback of this model is that the random variables that affect the probability of choosing bundles with overlapping products are assumed independent.

Immorlica et al. (2018) consider a choice model based on vertical customer differentiation, i.e., the ordering of utility of products is unambiguous and is the same for all customers, but customers differ in how much value they extract from the products. Unlike them, we study a *horizontally differentiated* choice model, with different customers having different utility maximizing sets. While Immorlica et al. (2018) focus on hardness results for the corresponding optimization problem, we restrict ourselves to bundles of size at most two for much of the paper (specializing the BundleMVL-K model family), and focus on designing scalable practical algorithms with extensive benchmarks, so that they can be readily used for near real-time personalization on e-commerce platforms.

4.2. The BundleMVL Choice Model For Multi-Purchase Behavior

The family of multi-purchase choice models that we formulate and study in this work, namely the BundleMVL-K family, has roots in the Marketing and the Spatial Statistics literature (Russell and Petersen 2000). Models in this family describe the probability with which a customer purchases a bundle S (with $|S| \leq K$) of unique products when the platform recommends set C of products. Thus, the suffix K parameterizes these models by the maximum size of bundles that a customer can purchase (for instance, BundleMVL-2 model captures purchases of bundles of size at most 2). We start by specifying the conditional utility (a conditional random variable) of purchasing a product given the purchase decisions corresponding to all other products that were offered for the generic BundleMVL-K model as:

$$U(i|\{X_j = x_j : j \in C, j \neq i\}) = \left(\alpha_i + \sum_{j \in C, j \neq i} \beta_{ij} x_j + \epsilon_i\right) \mathbb{1}\left\{\sum_{j \in C, j \neq i} x_j < K\right\}, \tag{4.1}$$

where α_i is a product specific parameter, parameters β_{ij} capture interactions between product pairs i and j, ϵ_i is a noise random variable distributed according to the Gumbel distribution, $\mathbb{1}\{\}$ is the indicator function that evaluates to one (zero) if the inequality is true (false), and X_j represent binary random variables that signify whether the customer purchased item j or not (x_j are the corresponding realizations) when C is offered. The β_{ij} parameters are symmetric in the product pair, i.e., $\beta_{ij} = \beta_{ji}$. We can interpret from the above equation that the conditional utility of adding a product to a purchased bundle depends on its intrinsic value and its pairwise relationship with other purchases. (Thus, even if the bundles are of size K, the effect on the utility by other products is pairwise, restricting the number of parameters to be of $O(n^2)$, where n is the number of products.) As an example, consider the following context. In grocery shopping, the utility of buying eggs for a customer who has already decided to buy flour and butter (which would let them make a cake) as part of this bundle can be higher, as compared to buying eggs when they had decided to not purchase flour and butter.

If the consumer has made a decision for each of the other products, then they will purchase product i only if the above conditional utility exceeds the threshold value 0. The conditional probability of buying product $i \in C$ can be computed as:

$$P\left(X_i = 1|\{X_j = x_j : j \in C, j \neq i\}\right) = \frac{\exp(\alpha_i + \sum_{j \in C, j \neq i} \beta_{ij} x_j)}{\exp(\alpha_i + \sum_{j \in C, j \neq i} \beta_{ij} x_j) + 1} \mathbb{1}\left\{\sum_{j \in C, j \neq i} x_j < K\right\}. \tag{4.2}$$

This form of the probability is due to the noise being Gumbel distributed. Also, note that the above probability is non-zero only when the number of products already purchased is strictly less than K. These conditional probabilities can be combined using Besag's characterization theorem (Besag 1974) to get a consistent joint probability distribution of purchase of bundles. Let ϕ be the empty bundle signifying a no-purchase event. As per Besag's theorem, for any $\mathbf{x} = (x_1, \cdots x_n)$ such that $P(\mathbf{x}) > 0$, we have $\frac{P(\mathbf{x})}{P(\phi)} = \prod_{i \in C} \frac{g^i(x_i)}{g^i(0)}$, where $g^i(1) + g^i(0) = 1$ and:

$$g^i(1) = \frac{\exp(\alpha_i + \sum_{j \in C, j<i} \beta_{ij} x_j)}{\exp(\alpha_i + \sum_{j \in C, j<i} \beta_{ij} x_j) + 1} \mathbb{1}\left\{ \sum_{j \in C, j<i} x_j < K \right\}.$$

Thus, $\frac{P(\mathbf{x})}{P(\phi)} = \exp\left(\sum_{i \in C} \alpha_i + \sum_{i \in C} \sum_{j \in C, j<i} \beta_{ij} x_j\right)$ if $\mathbb{1}\{\sum_{j \in C, j<i} x_j < K\} = 1$ and 0 otherwise. Using the fact that the sum of probability of purchase over all bundles is one, we get the probability of purchase of a bundle S given C as: $P(S) = \frac{V_S}{1 + \sum_{S' \subseteq C, |S'| \leq K} V_S'}$, where $V_S = \exp\left(\sum_{i \in C} \alpha_i x_i + \sum_{i \in C} \sum_{j \in C, j<i} \beta_{ij} x_i x_j\right)$ $\forall\ S \subseteq C,\ |S| \leq K$, and $x_j = 1$ if $j \in S$ and zero otherwise. This expression lends itself to a high degree of interpretability (positive β_{ij} implying complementary and the opposite substitution relations), and intuitively suggests that bundles with large positive values of the intrinsic parameters (α_is) and large positive pairwise affinity values (β_{ij}s) will have a higher probability of being purchased. Note that one can extend our model representation power by including parameters involving three or more products as well, because as long as we ensure that the conditional probability of purchasing an product does not depend on the order of previous purchases, Besag's theorem can still be used to derive an analogous multi-purchase model. To be consistent with the literature on single-choice models, we introduce another parameter v_0 corresponding to the no-purchase probability by scaling each V_S by $\frac{1}{v_0} V_S$. One can interpret this parameter as the result of comparing the conditional utilities to a non-zero threshold.

A key point to note is that the model is not derived based on assigning a mean utility to every bundle, adding a random noise term and deriving purchase probabilities based on the utility maximizing bundle. Instead, the model assumes a form on the conditional utility of purchase of a product given the purchase decision on all other products and derives the only joint distribution consistent with this condional probability distribution.

While precursors to the BundleMVL-K family exist in the literature, our key contribution here is two fold: (a) we extend these precursors to the modeling of multiple products in addition to categories (a category is a collection of similar products), and (b) we are able to

restrict the number of products purchased to a parameterized value K, and still obtain the *softmax* (or sigmoidal) structure of the probability of purchase expression above. This restriction parametrized by K and the softmax structure are both very appealing from an optimization perspective (see Section 4.4). For instance, the MNL single-choice model also has a similar form, which allows for the expected revenue maximizing set computation in polynomial time. We achieve these two properties without making a restrictive assumption that is made for many other random utility models in the literature (Benson et al. 2018): they assume independent Gumbel perturbations to the utilities of bundles, even if these bundles share products between them.

With increasing values of K, we can model customers that purchase larger bundles ($K = 1$ gives us the MNL model). The model parameterization (α, β) remains the same for $K \geq 2$, although the data likelihood changes. The choice of K is driven by the suitability of the model to data, and the tractability of the optimization problems (maximizing expected revenue or probability of purchase) downstream. For example, if most customers buy less than say 10 products, although choosing K as 10 in the BundleMVL-K model suffices (see Figure 4.1), it may still be prudent to trade-off model fit with the computational tractability of optimization (by choosing a smaller value of K). As we show in Section 4.5, the choice of $K = 2$ strikes a great balance for many real world datasets in terms of fit and in terms of scalability of personalization. In particular, we show that on one hand the revenue gains achieved by using the BundleMVL-2 model over the MNL model can be large especially as the number of products grow, and on the other hand optimizing over BundleMVL-2 is much more empirically tractable (more revenue gains in a small fraction of time) when compared to BundleMVL-3.

Any BundleMVL-K model can be estimated directly using maximum likelihood estimation (MLE). Let $\{S_l, C_l\}_{l=1}^m$ be the dataset that potentially includes no purchase observations and $\max_l |S_l| \leq K$. Then the likelihood of observing the given data is:

$$P_{data} = \prod_{l=1}^{m} P(S_l | C_l \text{ was offered}) = \prod_{l=1}^{m} \left(\frac{\exp\left(\sum_{i \in S_l} \alpha_i + \sum_{i,j \in S_l, i<j} \beta_{ij}\right)}{v_0 + \sum_{S' \subseteq C_l, |S'| \leq K} \exp\left(\sum_{i \in S'} \alpha_i + \sum_{i,j \in S', i<j} \beta_{ij}\right)} \right), \tag{4.3}$$

which can be maximized numerically using, say first order methods, to get estimates of α_is and β_{ij}s. The estimation problem becomes easy when $K = 2$ because one can optimize directly over V_S.

Lemma 4.2.1. Given values V_S for all bundles S of size ≤ 2, we can uniquely obtain the

parameters (α, β) by solving the system of equations: $\log V_S = \sum_{i \in S} \alpha_i + \sum_{i,j \in S, j>i} \beta_{ij}$ for all S.

The proof follows by showing that the linear transformation specified above between $\log V_S$ and α_i's and β_{ij}'s is invertible.

Further, if the data is such that the same recommendation set is shown to consumers in each observation, then V_S can be estimated by simply counting and normalizing, which is a linear time computation. Because of the equivalence above, we will use V_S as parameters in the rest of the paper when working with BundleMVL-2. For estimating the BundleMVL-K model from data with transactions containing bundles of size greater than K, we propose splitting such transactions into multiple transactions each of which has a bundle of size at most K. This transformation allows computing the likelihood of the original transactions, which enables direct likelihood comparisons across models. The detailed procedure is described in Appendix C.1.

Note that the BundleMVL-K model is equally applicable at an individual level as it is at a population level. In particular, in the presence of richer data at a user level that could include additional user features, personalized α, β values can be learned for each customer by direct segmentation or by learning a parametric function that outputs α, β given the user features(Feldman et al. (2019) gives an example of how personalized parameter values can be learned in a live e-commerce system for the MNL choice model). This paves the way for doing personalized recommendations on e-commerce websites.

If information about product features is also available, then the α and β parameters can be depend on these features as well. The mapping from such features to α, β values can be constrained by dimensionality/degrees-of-freedom so as to make it amenable to interpretability (such as being able to reason about what product features makes products complementary/substitutional).

A brief background on the MMC model: We use the MMC model (Benson et al. (2018)) later as a comparison benchmark. This model can be thought of as a two-stage mixture model. In the first step, the customer chooses the size of the bundle of products they are going to purchase. The size of this bundle is m with probability proportional to z_m for $1 \leq m \leq |C|$ where C is the recommendation set. In the second step, the customer chooses an m-sized bundle S by assigning random utilities $\log V_S + \epsilon_S$. Here $\log V_S = \sum_{i \in S} \hat{\alpha}_i + \hat{\beta}_S$, parameters $\hat{\alpha}_i$ represent intrinsic utility of product i, $\hat{\beta}_S$ represents additional utility from buying the products together as a bundle S and ϵ_S's are i.i.d. Gumbel distributed random variables associated with every bundle

S. The parameters $\hat{\beta}_S$ take non-zero values only for a small number of bundles S, leading to a sparse representation. To incorporate the possibility of not making any purchase, the model can be extended to consider a utility of not making any purchase as $\log V_0^m + \epsilon_\phi^m$. Thus, the probability of choosing a bundle S of size m when a recommendation set C is shown is given by $\frac{z_m}{z_1+\cdots z_m} \frac{V_S}{V_0^m + \sum_{S' \subset C s.t. |S'|=m} V_{S'}}$. The specific structure of the underlying generative process (i.e., the two stages described above) leads to computational intractability of estimation, which is avoided in our case.

4.3. Revenue Maximization: Hardness and Structural Results

Maximizing revenue from a set of offered products is one of the key goals of online/offline retailers. Under the BundleMVL-K model, the expected revenue of the platform if it offers recommendations C is given by $R_K(C) = \frac{\sum_{S \subseteq C, |S| \leq K} V_S r_S}{v_0 + \sum_{S \subseteq C, |S| \leq K} V_S}$, where $r_S = \sum_{i \in S} r_i$, and $r_i > 0$ is the fixed and known revenue corresponding to product $i \in W$ and $W = \{1, 2, \cdots n\}$ is the set of all products. We assume that the products are ordered such that $r_1 \geq r_2 \cdots \geq r_n$. As mentioned earlier, we will primarily focus on the BundleMVL-2 model, whose expected revenue function can be written more explicitly as:

$$R_2(C) = \frac{\sum_{i \in C} r_i V_{\{i\}} + \sum_{i \in C} \sum_{j \in C, j > i} (r_i + r_j) V_{\{i,j\}}}{v_0 + \sum_{i \in C} V_{\{i\}} + \sum_{i \in C} \sum_{j \in C, j > i} V_{\{i,j\}}} = \frac{\sum_{i \in W} \sum_{j \in W} \hat{r}_{ij} \theta_{ij} x_i^C x_j^C}{v_0 + \sum_{i \in W} \sum_{j \in W} \theta_{ij} x_i^C x_j^C}, \tag{4.4}$$

where $x_i^C = 1$ if $i \in C$ else 0, $\theta_{ij} = \begin{cases} \frac{V_{\{ij\}}}{2} & i \neq j \\ V_{\{i\}} & i = j \end{cases}$, and $\hat{r}_{ij} = \begin{cases} r_i + r_j & i \neq j \\ r_i & i = j \end{cases}$. Our objective is to find a recommendation set C^* that has the maximum expected revenue among all feasible recommendation sets (represented using collection $\mathcal{C}$):

$$\max_{C \in \mathcal{C}} R_2(C). \tag{4.5}$$

We first consider the unconstrained case, i.e., when $\mathcal{C} = 2^W$. We show that this problem is NP-complete, and consequently derive structural properties of the optimal solution that will be used effectively in algorithm design (Section 4.4.1). The case when the collection of feasible recommendation sets is given by linear constraints is tackled in Section 4.4.2.

4.3.1. Hardness of Optimization

The revenue functions $R_2(C)$ and $R_K(C)$ have a form similar to the expected revenue of recommendations under the MNL model. For the MNL model, it is known that the unconstrained revenue optimization problem can be solved in linear time as the optimal set is a revenue-ordered set, i.e., it only contains the l highest priced products for some $l \in \mathbb{Z}_+$. But for the BundleMVL-2 model, we show that this does not hold.

Theorem 4.3.1. [Hardness Result for BundleMVL-2] Under the BundleMVL-2 model, the decision version of the unconstrained revenue optimization problem (4.5) is NP-complete.

The proof of the theorem follows from a reduction of the well-known MAXCUT problem and is given in the appendix. While the existence of a polynomial time approximation algorithm for problem (4.5) is an open question, we believe it is inapproximable because it is similar to the quadratic knapsack problem with additional constraints on the coefficients (that can be both positive and negative). And it is known that the quadratic knapsack problem (defined on an edge-series parallel graph) is hard to approximate (Rader Jr and Woeginger 2002). On a related note, we also provide a similar hardness result for the MMC multi-choice model (proof is in the appendix).

Theorem 4.3.2. [Hardness Result for MMC] The decision version of the unconstrained revenue optimization problem under the MMC model (with number of allowed purchases ≤ 2) is NP-complete.

These results are not surprising given similar results in prior works for single-choice models (Rusmevichientong et al. 2010b). In the absence of polynomial time exact algorithms for the optimization problem (4.5), we can either develop exact algorithms without any run-time guarantees, or create heuristic polynomial time algorithms. We explore both these directions, and improve them based on new structural properties, which we describe next.

4.3.2. Structural Properties of the Optimal Solution

Under the setting $\mathcal{C} = 2^W$, i.e., for the unconstrained optimization problem (4.5), we prove useful structural properties satisfied by the optimal solution C^* (proofs of these results are in the appendix):

Lemma 4.3.3. For all products $i \in W$ that are not in any optimal recommendation set C^*, $r_i \leq R_2(C^*)$. Equivalently, $C_u^* \subseteq C^*$, where $C_u^* = \{i : r_i > R_2(C^*)\}$.

Lemma 4.3.4. Let C^* be an optimal recommendation set. For every $i \in C^*$, $\exists\ j(i) \in C^*$, where $j(i) \neq i$ and $r_i + r_{j(i)} \geq R_2(C^*)$.

Remark: If C^* is an optimal recommendation set of the smallest cardinality, then $\forall\ i \in C^*$, $\exists\ j(i) \in C^*$, where $j(i) \neq i$ and $r_i + r_{j(i)} > R_2(C^*)$.

Lemma 4.3.5. Let the i-th revenue-ordered recommendation set be defined as $A_i = \{1, 2, \cdots i\}, i \in W$. Then, the revenue of revenue-ordered recommendation sets increases monotonically as long as the price of all the products in the revenue-ordered recommendation set is greater than $R_2(C^*)$, i.e., $R_2(A_1) \leq R_2(A_2) \cdots \leq R_2(A_k)$ where $r_k > R_2(C^*) \geq r_{k+1}$.

Lemma 4.3.3 says that if a product's revenue is greater than the optimal revenue, then it belongs to the optimal recommendation set. Lemma 4.3.4 suggests that a product that is in the optimal recommendation set has a corresponding pair that also belongs to the optimal set such that the sum of their revenues is greater than the expected revenue of the set. Finally, Lemma 4.3.5 suggests that the objective function has a partial monotonicity property. Overall, these three properties can help us narrow the search for the optimal recommendations. For instance, if an algorithm keeps an estimate of an upper bound on the optimal revenue, then this can help prune the search space based on Lemma 4.3.3.

4.4. Algorithms for Revenue Maximizing Recommendations

We propose a new algorithmic approach for solving problem (4.5). We first consider the unconstrained problem i.e. when $\mathcal{C} = 2^W$ and then consider the setting with linear constraints. Our approach is based on binary search and outputs recommendations having revenue within a specified range of the optimal revenue. The iterative nature of this approach helps in terminating the search for the optimal recommendations gracefully, especially under stringent timing requirements expected in *personalization focused* applications on e-commerce platforms. We also consider three benchmark methods for the same optimization problem - one of which gives the optimal solution and the other two are heuristic. We discuss structural properties of the solutions obtained by the heuristic algorithms and examine conditions under which they can be optimal.

4.4.1. Binary Search with Efficient Comparisons

A binary search based efficient algorithm for a single-choice model was first described in Sinha and Tulabandhula (2020), who evaluate the efficacy of using nearest neighbor search techniques for capturing arbitrary feasibility constraints in the comparison step. Building on their algorithmic strategy, we devise a binary search outer loop and focus on improving the computational speed of the comparison steps by *exploiting the structure of our multi-purchase choice model*. While such a use of the binary search template is common for transforming optimization problems to feasibility problems (and vice versa), we show below that in our context this allows for two key efficiency gains, namely: (a) use of structural properties reduces search space significantly, and (b) allows for extensively developed heuristics for the comparison step giving us scalability. For any given tolerance $\epsilon > 0$, this algorithm gives an ϵ-optimal solution, i.e., a solution within ϵ of the optimal value. In each iteration of the search process, we narrow the size of the interval in which $R(C^*)$ lies as outlined in BinarySearchAO (Alg. 3). (The upper bound on $R(C^*)$ is initialized as the maximum revenue possible from a bundle of two products i.e. $r_1 + r_2$. The optimal assortment is arbitrarily initialized as $\{1\}$ and is of relevance only when the optimal revenue is less than ϵ, in which case all asssortments have revenue within ϵ of the optimal assortment.

The computationally expensive comparison step (Compare-Step, line 4 in Alg. 3) checks if the optimal revenue is greater than the specified threshold κ_j. A key insight is that this calculation can be transformed as follows:

$$\max_{C\in\mathcal{C}} R(C) \geq \kappa_j \iff \max_{C\in\mathcal{C}} \sum_{i\in W}\sum_{j\in W} \theta_{ij} x_i^C x_j^C (\hat{r}_{ij} - \kappa_j) \geq \kappa_j v_0. \tag{4.6}$$

The optimization problem on the left hand side of the transformed comparison is a quadratic integer program (QIP). In the absence of constraints (e.g., capacity constraints), this problem is a member of a specialized class of QIP problems known as Quadratic Unconstrained Binary Optimization (QUBO) problems, which are known to be NP-hard (Pardalos and Jha 1992). Although the set of all comparison step (QIP) instances is a strict subset of the QUBO instances, for simplicity we will refer to the comparison step instance as a QUBO instance when the context is clear. The binary search template can also be used to solve other BundleMVL-K models (the Compare-Step will be different), and Section 4.5 documents its use with the BundleMVL-3 model.

Algorithm 3 BINARYSEARCHAO

Require: Parameters $\{r_i\}_{i=1}^n$, $\{\theta_{ij}\}_{i=1,j=1}^n$, tolerance level $\epsilon > 0$, and feasible sets $\mathcal{C}$.

$L_1 = 0, U_1 = r_1 + r_2, j = 1$, and $C^* = \{1\}$.
while $U_j - L_j > \epsilon$ **do**
  $\kappa_j = (L_j + U_j)/2$.
  if $\kappa_j \leq \max_{C \in \mathcal{C}} R(C)$ **then**
    $L_{j+1} = \kappa_j, U_{j+1} = U_j$.
    Pick any $C^* \in \{C \in \mathcal{C} : R(C) \geq \kappa_j\}$.
  else
    $L_{j+1} = L_j, U_{j+1} = \kappa_j$.
  Increment j by 1.
return C^*

Algorithm 4
BINARYSEARCHAO(EFFICIENT)

Require: Parameters $\{r_i\}_{i=1}^n$, $\{\theta_{ij}\}_{i=1,j=1}^n$, tolerance level $\epsilon > 0$, and feasible sets $\mathcal{C}$.

$U_1 = r_1 + r_2, j = 1, i = 1$ and $C^* = \{1\}$.
while $r_{i+1} \geq r_i$ **do** Increment i by 1.
$L_1 = r_{i+1}$.
while $U_j - L_j > \epsilon$ **do**
  $\kappa_j = (L_j + U_j)/2$.
  if $\kappa_j \leq \max_{C \in \mathcal{C}} R(C)$ **then**
    $L_{j+1} = \kappa_j, U_{j+1} = U_j$.
    Pick a C^* such that $R(C^*) \geq \kappa_j, \overline{B} \subset C^*$, and $\underline{B} \cap C^* = \phi$; where $\overline{B} = \{i : r_i > U\}$, and $\underline{B} = \{i : r_i + r_1 < L\}$.
  else
    $L_{j+1} = L_j, U_{j+1} = \kappa_j$.
  Increment j by 1.
return C^*

Using Structural Properties of C^*:

BinarySearchAO can be made more efficient by using the properties of the optimal recommendation set derived earlier (Lemma 4.3.3-4.3.5). At the cost of additional pre-processing (which is small), we can start with a lower bound greater than 0 based on Lemma 4.3.5. Let l be the maximum index i such that $R(A_1) \leq R(A_2) \cdots \leq R(A_i)$. Then, from Lemma 4.3.5, we know that $l \geq k$ (see the definition of k in the Lemma). Thus, $r_k \geq R(C^*) \geq r_{k+1} \geq r_{l+1}$. Hence, at the beginning of the binary search, the lower bound L can be set as r_{l+1}.

Lemmas 4.3.3 and 4.3.4 can be used to make the comparison step faster. In particular, when we have an upper bound U on $R(C^*)$, then using Lemma 4.3.3, we know that all products with revenue greater than U should belong to the optimal recommendation set. Similarly, with a lower bound L on the revenue of the optimal recommendation set, we know that all products i such that $r_i + r_1 < L$, cannot belong to the optimal recommendation set. These observations predetermine the fate of some products, reducing the problem size (sometimes significantly as seen in our experiments) in the comparison step (4.6). BinarySearchAO(Efficient) incorporates these properties is shown in Algorithm 4.

Solving the QUBO Problem Approximately:

Though the QUBO problem is an NP-hard problem (Pardalos and Jha 1992) as discussed before, there has been ample research in heuristic algorithms that return high quality solutions in extremely reasonable computation times (Dunning et al. 2018). This makes it appealing to use these approximate algorithms in solving the Compare-Step. Nonetheless, solving problem (4.6) approximately can potentially lead to narrowing down on an incorrect interval in the binary search outer loop. We take two steps to alleviate this issue:

Firstly, for each QUBO problem, we run multiple QUBO heuristics in parallel, as seen in our experiments (Section 4.5). The binary search interval will have an incorrect update only if all the heuristics lead to an incorrect answer for the Compare-Step. Secondly, we further robustify the binary search outer loop by using a noisy binary search variant (Burnashev and Zigangirov 1974). Here one maintains a distribution over the unknown $R(C^*)$, and each comparison (with the median of the current distribution) is used to obtain an updated distribution using Bayes rule. This prevents incorrect comparison step outcomes from easily misleading the search process. For any specified tolerance level and a probability value with which the given solution needs to lie within the tolerance level, the number of iterations required for the noisy binary search still

stays logarithmic. We refer to this algorithm as NoisyBinarySearchAO and its version which uses the structural properties as NoisyBinarySearchAO(Efficient) (we omit its description due to space constraints).

4.4.2. Optimization with Linear Constraints

Constraints on the feasible recommendation sets are common in practice. For example, there can be a *cardinality* constraint on the maximum number of products in a recommendation set due to webpage/screen size limits in the online e-commerce setting, or due to limited shelf space in the offline retail setting. Business rules and obligations such as ensuring sufficient representation from various sub-groups of products, or the requirement to maintain a precedence order among products can also be formulated as linear constraints (Davis et al. 2013, Sinha and Tulabandhula 2020). The binary search algorithm (Algorithm 3) can incorporate linear constraints in the following way. For a general linear constraint set $Dx = e$, the Compare-Step becomes

$$\max_{\{x\in\{0,1\}^n : Dx=e\}} \sum_{i\in W}\sum_{j\in W} \theta_{ij}x_i x_j(\hat{r}_{ij}-\kappa) \geq \kappa v_0.$$

This is a quadratic binary optimization problem with constraints that can be relaxed to get

$$\max_{x\in\{0,1\}^n} \sum_{i\in W}\sum_{j\in W} \theta_{ij}x_i x_j(\hat{r}_{ij}-\kappa) + \lambda(Dx-e)'(Dx-e),$$

for a suitably large $\lambda < 0$. In this form, the aforementioned QUBO solvers can be used directly.

If we have an inequality constraint, then as long as the components of D and e are integral, a similar transformation can be done with an appropriate number of slack variables (Glover and Kochenberger 2018). For instance, suppose we want to ensure that the size of the recommendation set is at most $e \in \mathbb{Z}_+$, i.e., we have the constraint $\mathbf{1}'x \leq e$ (here, $\mathbf{1}$ is the vector of all ones). Then, using slack variables $s_1, ..., s_e$, we get the following readily solvable QUBO instance:

$$\max_{x\in\{0,1\}^n} \sum_{i\in W}\sum_{j\in W} \theta_{ij}x_i x_j(\hat{r}_{ij}-\kappa) + \lambda(\mathbf{1}'x + \sum_{i=1}^{e} s_i - e)'(\mathbf{1}'x + \sum_{i=1}^{e} s_i - e). \tag{4.7}$$

4.4.3. Benchmark Algorithms

We briefly describe three benchmark algorithms we considered - one of which is an exact algorithm, and the other two are heuristics extending exact algorithms for some single-choice models.

The first benchmark is based on a mixed integer program (MIP) formulation and gives an exact solution. This formulation builds on an earlier formulation for the *mixture of MNLs* model studied in Blanchet et al. (2016), and is described in Algorithm 9 in the appendix. This benchmark can also easily incorporate linear constraints mentioned above.

Our next benchmark is a generalization of the ADXOPT algorithm Jagabathula (2014), called the ADXOPTL algorithm (Alg. 10 in the appendix, L is a parameter). This is a choice model agnostic greedy algorithm which in every iteration looks for a set of L products whose addition/deletion/exchange will lead to recommendations with a higher revenue. The solution $\widehat{C}$ of ADXOPT2 (i.e., L=2) is guaranteed to contain all relevant *high-revenue* products as per the below lemma (proof in appendix).

Lemma 4.4.1. Let $\widehat{C}$ be the solution returned by ADXOPT2 using the BundleMVL-2 model. Then, $\widehat{C} \supset C_u^*$, where $C_u^* = \{i : r_i > R_2(C^*)\}$.

The (worst-case) time complexity of the ADXOPT2 algorithm is $O(n^7)$, which is prohibitive for medium to large instances as observed in our experiments.

Our third benchmark is a heuristic algorithm which chooses the revenue-ordered recommendation set with the highest revenue, and has a time complexity of $O(n^3)$. This heuristic is known to be optimal or close to optimal for several single-choice models (Rusmevichientong et al. 2010b). In what follows, we characterize conditions under which this heuristic performs well for the BundleMVL-2 model, by generalizing the notion of *value conscious* customers introduced in Rusmevichientong et al. (2014) to our multi-purchase setting. These customers prefer a cheaper product (or pair of products) when compared to a more expensive product (or pair of products) but derive more value from a higher priced product (or pair of products). Here value is defined as the product of the revenue and the utility parameters associated with the set of purchased items. This intuition is formalized below:

Assumption 4.4.2. Model parameters for value conscious customers satisfy:

1. $V_{\{i\}} \leq V_{\{j\}}$ and $V_{\{i,k\}} \leq V_{\{j,k\}}$ $\forall i < j, i \neq k, j \neq k$ and $i, j, k \in W$
2. $r_i V_{\{i\}} \geq r_j V_{\{j\}}$ and $(r_i + r_k) V_{\{i,k\}} \geq (r_j + r_k) V_{\{j,k\}}$ $\forall i < j, i \neq k, j \neq k$ and $i, j, k \in W$.

Theorem 4.4.3. For value conscious customers, the revenue-ordered heuristic produces an optimal recommendation set for the optimization problem $\max_{|C| \leq d} R_2(C)$ for any $d > 0$.

When all the $V_{\{i,j\}}$ values are 0, the BundleMVL-2 model reduces to an MNL model, and the optimal recommendation set is a revenue-ordered recommendation set for the unconstrained optimization problem. This gives us another set of conditions where we can show that the revenue-ordered heuristic produces good solutions, namely when the $V_{\{i,j\}}$ values are small compared to $V_{\{i\}}$ values:

Assumption 4.4.4. Model parameters satisfy: $\max_{i,j\in W, i\neq j} V_{\{i,j\}} \leq \epsilon \min_{k\in W\cup\phi} V_{\{k\}}$.

Theorem 4.4.5. Under Assumption 4.4.4, the revenue-ordered heuristic satisfies: $R_2(C^*_{revord}) \geq \frac{2-\epsilon|C^*_{MNL}|}{2+4\epsilon|C^*|} R_2(C^*)$, where $C^*_{revord} \in \arg\max_{C\in\{A_1,A_2\cdots A_n\}} R_2(C)$, and C^*_{MNL} and C^* are the optimal solutions of the unconstrained problem under the MNL and the BundleMVL-2 models respectively.

We note in passing that guarantees similar to the above can be obtained for BINARY-SEARCHAO(EFFICIENT) if we can obtain an approximation guarantee for the COMPARE-STEP under the same assumptions. We omit this analysis here for brevity, and shift out attention to evaluating the value of modeling multiple purchases and of our algorithms using real datasets next.

4.5. Experiments

We perform two sets of experiments. The goal in the first set is to validate the merits of the BundleMVL-2 model compared to other models on real data based on the empirical fit and the revenue obtained by optimizing based on this model. In the second set, we benchmark the solution quality and computational times of the optimization approaches (Section 4.4) in detail, to ascertain their suitability in offline and online recommendation settings.

All results reported here are based on 50 Monte Carlo runs for each configuration, unless otherwise noted. For computation times and the optimality gap metrics, we have plotted the median along with the 25th and 75th percentiles. We segment the number of products n into three regimens, viz., (a) Small: $20-80$, (b) Medium: $100-400$, and (c) Large: $500-1500$ products, while discussing scalability trends. Additional runs/variations of these experiments are documented in the appendix.

Dataset	Bakery	WalmartItems	WalmartDepts	Kosarak	Instacart	LastFMGenres	ycItems	ycDepts	Ta Feng	UCI Online Retail
No. of products	50	1075	66	2621	5981	443	22915	31	3357	3350
No. of unique purchased bundles	1267	3895	1424	24921	100238	9866	41127	101329	68597	7739
No. of observations	17171	25135	73780	286399	298332	471638	208049	1915	95001	11056
No. of unique recommended sets	1	1	1	1	1	1	160711	1915	1	1

Table 4.2: Summary of the datasets used for comparing empirical fit of different models.

4.5.1. Suitability of the BundleMVL-2 Model on Real Data

We perform three assessments: (a) the fit of the BundleMVL-2 model on real world datasets, (b) qualititative insights from the estimated parameters of the BundleMVL-2 model, and (c) the revenue gains that can be achieved with this model compared to other competing models.

Empirical Fit of BundleMVL-2 vs Others:

We compare the BundleMVL-2 model with the MMC and the MNL models. For this, we use five datasets (Bakery, Walmart, Kosarak, Instacart, LastFMGenres) that have fixed recommendation sets across all interactions, and one dataset from YOOCHOOSE (two variants ycItems and ycDepts) that has variable recommendation sets (see Benson et al. (2018) for their descriptions). While some of these are not explicitly about purchases, they do capture multi-choice behavior of consumers. Additionally, we also use the Ta Feng Grocery (https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset) and the UCI Online Retail (https://archive.ics.uci.edu/ml/datasets/online+retail) datasets that contain information about the revenues/prices in addition to the bundles purchased in the subsequent sections. These datasets are from a variety of domains, and Table 4.2 summarizes the relevant characteristics. We remove extremely infrequent products ($\sim 10\%$ on average) from the datasets. Further, when estimating a BundleMVL-K model, we transform all observations into ones that involve at most K-sized bundle purchases as described in Appendix C.1.

All datasets are split in the ratio $80:20$, with the former used for learning the parameters, and the latter for reporting out-of-sample log-likelihood fit. As these datasets do not contain information about customers seeing the recommendation set but leaving without making any purchase, it is not possible to estimate v_0. Consequently, we rely on domain knowledge to pick an appropriate value in our experiments although this has been observed to vary wildly from one application to another. While the BundleMVL-2 and MNL have no tunable parameters, we need to pick the number of *corrections* for the MMC model. Increasing the number of corrections increases the flexibility of this model, but also increases the number of parameters. While adding such corrections, we pick bundles (namely the *H-sets*) in descending order of their frequency of

Dataset	Walmart Items Dataset			Walmart Depts Dataset		
Model	**#parameters**	**train_ll**	**test_ll**	**#parameters**	**train_ll**	**test_ll**
MNL Model	1075	-160299	-40526	66	-302881	-76461
MMC model (0 corrections)	1075	-168250	-42494	66	-326263	-82210
MMC model (1% corrections)	1098	-142794	-35985	78	-317403	-79961
MMC model (5% corrections)	1194	-138402	-35017	130	-308318	-77761
MMC model (20% corrections)	1551	-135825	-34871	323	-303441	-76557
MMC model (50% corrections)	2265	-131954	-35018	709	-301738	-76208
MMC model (100% corrections)	3456	-125531	-48916	1353	-300710	-78222
BundleMVL-2 model	3384	-125410	**-26518**	1289	-276752	**-69024**

Table 4.3: Log-likelihood values under different models for the Walmart dataset.

appearance in the datasets (this heuristic is provided in Benson et al. (2018)).

The fit of the three models are tabulated for the two Walmart datasets (also see Figure 4.1 in Section 4.1) in Table 4.3 and for the other six datasets (omitting Ta Feng and UCI) in Table C.1 in the appendix. The number of corrections in MMC model have been reported as a percentage (between 0% to 100%) of the number of unique subsets of purchases of size greater than one that are observed. The column *#parameters* denotes the number of non-zero parameters in each model. The number of parameters in BundleMVL-2 differs from MMC with 100% corrections because of the way these are estimated, which leads to different numbers of parameters ending up non-zero in each case. The columns *train_ll* and *test_ll* contain the train and test log likelihood values respectively. We fit the MNL and BundleMVL-2 model based on MLE. For the case where the recommendation sets are the same across observations, the parameter estimates can be obtained analytically. In the general case, we use stochastic gradient descent with reasonable defaults. From these two tables, we infer that the BundleMVL-2 model provides the best fit (i.e., the highest log-likelihood value) in seven of the eight datasets, when compared to the MNL model and the MMC models with different levels of corrections. The gap between the BundleMVL-2 and MMC models when compared to the single-choice MNL model is quite large (up to $1.5-2\times$ lower log-likelihoods for the latter in some cases), validating the necessity for multi-purchase choice modeling. In summary, for all eight datasets, the BundleMVL-2 model is shown to be a viable and strong alternative to competitors (e.g., the MMC model) in capturing the rich multi-purchase behavior exhibited.

Qualitative Insights from Estimated BundleMVL-2 Model:

As shown above, the BundleMVL-2 model provides a good fit on multiple datasets that were in consideration. The estimated parameters of the model can give insights about the purchase patterns of products. In particular, the parameter β_{ij} for a pair of products i, j can signify the

additional utility due to buying these products together as a bundle. To validate this, we look at the product pairs with the highest β_{ij} values in the UCI dataset, which has product names available for many products. Based on our exploratory analysis, product pairs that have high β_{ij} values in the estimated BundleMVL-2 model include: (a) necklaces and earrings, (b) necklaces and bracelets, (c) placemats and coasters, and (d) wall art for gents and wall art for ladies, among others. These pairs of products are clearly complementary. The estimated BundleMVL-2 model thus, is able to learn the complementary relation between these pairs of products, and our optimization schemes would be able to use this information effectively for recommendation set optimization. A similar analysis can be done for product pairs with negative β_{ij} values. In addition to the product pairs which are clearly complementary/substitutable, we also see other pairs of products having high β_{ij}s. For instance, necklaces in two different colors, key rings with two different letters etched, numbered tiles with consecutive numbers, and candle plates and candle holders don't seem complementary at a first glance but still end up having high β_{ij} values. Some possible reasons for simultaneous purchase of two such similar items could be: a) if a customer has strong affinity for a product, they might want to have multiple versions of it (such as in different colors), or b) customers might want to purchase two similar items with the intention of choosing one among the two later on and eventually returning one of the products. In either case, information about these less obvious product pairs is also valuable for the retailer.

Revenues and Run-times with BundleMVL-2:

Even though the BundleMVL-2 model provides a good fit on real data, one of the primary goals of using such choice models in practice is to realize higher expected revenues, and this is our focus next. To start, we estimate parameters of the competing models using the Ta Feng and the UCI datasets. Next, we randomly sample (without replacement) a subset of products of pre-specified sizes, and define a choice model instance using their estimated parameter values. As there is no information about customer interactions that did not lead to a sale in these datasets, we fix the probability of no-purchase when all products are displayed to 30% (changing this probability to much higher and lower values while experimenting did not qualitatively change our conclusions), and estimate the v_0 parameter accordingly for each instance. We perform three comparisons: (1) revenue of BundleMVL-2 vs MNL, (2) revenue and run-times of BundleMVL-2 vs BundleMVL-3, and (3) run-times of BundleMVL-2 vs MMC. When doing these comparisons,

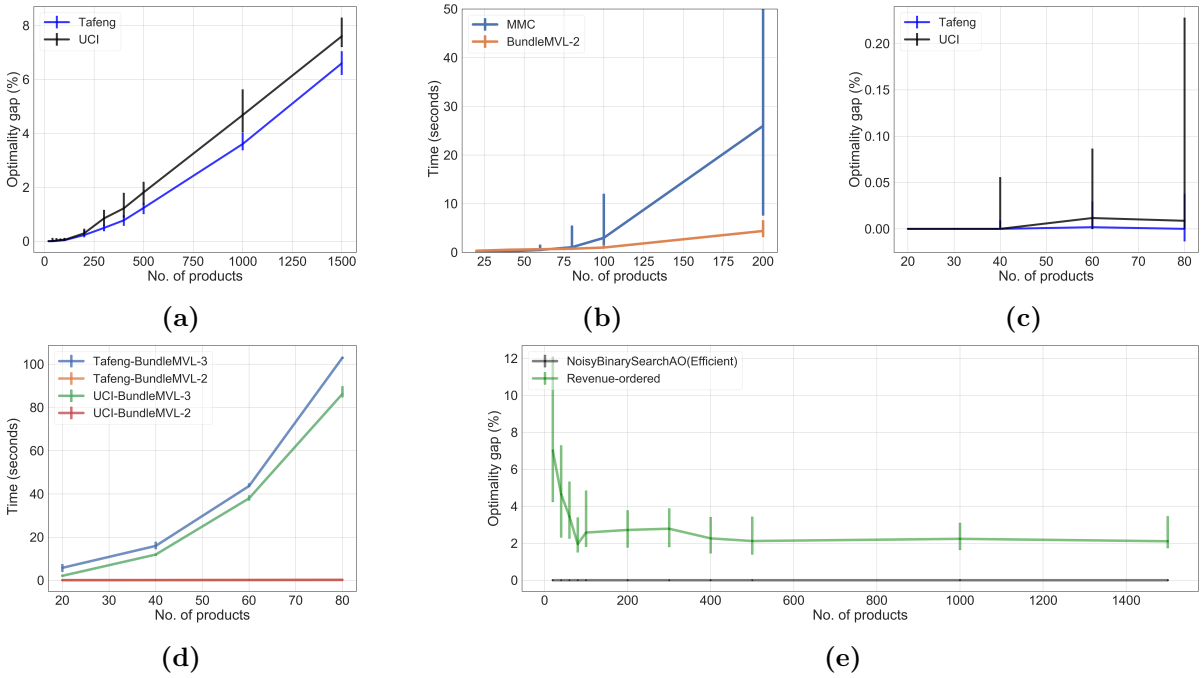


Figure 4.2: Performance plots. ((a)): Optimality gap of the optimal recommendation set as per the MNL model with the ground truth as the BundleMVL-2 model. ((b)): Time taken to solve for the optimal recommendation set for the unconstrained optimization problem under different models. ((c)): Optimality gap of the optimal recommendation set under the BundleMVL-2 model with the BundleMVL-3 model as the ground truth. ((d)): Time taken to solve the unconstrained optimization problem under different models. Ta Feng-BundleMVL-2 and UCI-BundleMVL-2 run-times overlap. ((e)): Assesing optimality of revenue-ordered heuristic.

we also study their trends as a function of the instance size (i.e., number of products n).

Comparison (1): Revenue of BundleMVL-2 vs MNL: Since the MNL model is a special case of the BundleMVL-2 model, we take the BundleMVL-2 model to be the ground truth, and calculate final revenues based on this. The optimality gap (which is the percentage difference from the optimal under ground truth) of the optimal assortment under the MNL model is shown in Figure 4.2a. Here, we observe that this optimality gap increases as the number of products increase. Moreover, it becomes significant for large product sizes (6-8% for 1500 products). Equivalently, a retailer can potentially increase their average revenue from a transaction by 6.4 - 8.7% in relative terms by moving from the single-choice MNL model to the BundleMVL-2 model. Thus, modeling purchase of multiple products (for instance, by using BundleMVL-2) can lead to significant revenue gains if we can also improve run-times of these models. For instance, since optimization over MNL is linear time, it is important to bring down the computational complexity of optimization over BundleMVL-2 model, and we investigate this exhaustively in Section 4.5.2.

Comparison (2): Revenue and run-times of BundleMVL-2 vs BundleMVL-3: Previously, we

made a remark that a significant portion of observed bundles are of size at most two in many real datasets and that the optimization problem becomes more challenging as K increases. Here, we compare if the potential revenue gains by using a richer model can trade-off the increase in computation. The setup is similar to the previous comparison. Here, the revenues are reported using the estimated BundleMVL-3 model as the ground truth model. In order to optimize over BundleMVL-3 model, we use a mixed non-linear integer programming solver called `Bonmin` from COIN-OR (Bonmin 2019). BinarySearchAO algorithm is used to solve for the optimal recommendation set under both models. Figure 4.2c shows the optimality gap of the BundleMVL-2 model. These optimality gaps are much smaller than the gaps observed in the BundleMVL-2-MNL comparison. Further, the time taken to perform the optimization increases dramatically for BundleMVL-3 as seen in Figure 4.2d. This likely stems from the difficulty in solving cubic integer optimization problems as compared to quadratic integer optimization problems. Thus, the revenue gains are small and do not outweigh the computational burden of optimizing over the BundleMVL-3 (and other BundleMVL-K models for $K > 3$) for large scale applications.

Comparison (3): Run-times of BundleMVL-2 vs MMC: A direct comparison of the revenues obtained using the BundleMVL-2 and MMC models is difficult because neither of the models are nested within the other. Moreover, a reasonable parsimonious multi-choice model that has both these models as special cases is hard to construct. Thus, we restrict our attention to comparison of run-times for computing the optimal recommendation set. To keep the comparison fair, we solve the corresponding integer programming formulation (see appendix) for each setting. Based on Figure 4.2b, we can conclude that the run-time is much faster under the BundleMVL-2 model, making it a better candidate for large-scale/time-constrained applications. Moreover, we demonstrate in the next subsection that solutions for the BundleMVL-2 model can be obtained even faster using the proposed binary search algorithms.

4.5.2. Run-times for Computing BundleMVL-2 based Recommendation Sets

We now focus on the BundleMVL-2 model and benchmark the computation times and quality of solutions produced by various algorithms described in Section 4.4 on the Ta Feng and UCI datasets. Similar evaluations for the MMC model is presented in the appendix. To get problem instances of different sizes, we use a similar procedure as described before wherein we learn parameters for the full dataset and then subsample the desired number of products and their

Heuristic	LAGUNA2009HCE	BURER2002	DUARTE2005	FESTA2002VNS
Ranked First	22.3%	20.2%	15.9%	14.8 %

Table 4.4: Performance analysis of QUBO heuristics: fraction of times the top heursitics gave the best solution

respective parameters. For the following list of experiments designed to illustrate how scalable our proposed algorithmic approaches are, we report results for the Ta Feng dataset and relegate results for the UCI dataset (which has similar trends as Ta Feng) to the appendix:

1. Selecting QUBO heuristics for the BundleMVL-2 model,
2. Benchmarking algorithms for the BundleMVL-2 model (unconstrained setting),
3. Benchmarking algorithms for the BundleMVL-2 model (constrained setting), and
4. Assessing optimality of the revenue ordered heuristic for the BundleMVL-2 model.

1. Selecting QUBO heuristics for the BundleMVL-2 model: As discussed in Section 4.4, we can use efficient approximate algorithms to solve the QUBO problem in conjunction with a (noisy) binary search outer loop to accelerate the search. Many heuristics have been proposed in the literature for solving the QUBO problem (Boros et al. 2007). We evaluate these heuristics to decide which ones to use in the NoisyBinarySearchAO algorithm.

To start with, we consider ∼20 heuristics discussed in Dunning et al. (2018) (`https://github.com/MQLib/MQLib`), and solve several QUBO instances of the Compare-Step stemming from optimizing over synthetically generated BundleMVL-2 models. Results show that none of the heuristics consistently outperform others in terms of solution quality. Thus, we measure the performance of each heuristic in terms of the proportion of times it gave the best solution among all heuristics. Table 4.4 summarizes the performance of the top four heuristics ranked in this manner. For subsequent experiments, we run all these top four heuristics at each comparison step of NoisyBinarySearchAO, and then select the best solution to resolve the comparison more accurately. Since the heuristics can be run in parallel, there is no significant increase in time taken as compared to using a single heuristic.

2. Benchmarking algorithms for the BundleMVL-2 model (unconstrained setting): Next, we benchmark our proposed and other competing approaches when there are no constraints on the feasible recommendation sets. Table 4.5 summarizes the list of algorithms that we consider in various product size regimes. The regimes in which each algorithm can be tested is decided based on the run-time and memory requirements of the algorithm. The MIP formulations are solved

Optimization Algorithm	Choice Model(s)	Type	Product Size Regime
BinarySearchAO	BundleMVL-2	ϵ-optimal	Small, Medium, Large
NoisyBinarySearchAO	BundleMVL-2	Heuristic	Small, Medium, Large
Mixed Integer Programming	BundleMVL-2/MMC	Exact	Small
Revenue Ordered	BundleMVL-2/MMC	Heuristic	Small, Medium, Large
Adxopt/Adxopt2	BundleMVL-2/MMC	Heuristic	Small

Table 4.5: Summary of optimization algorithms and the underlying multi-choice models.

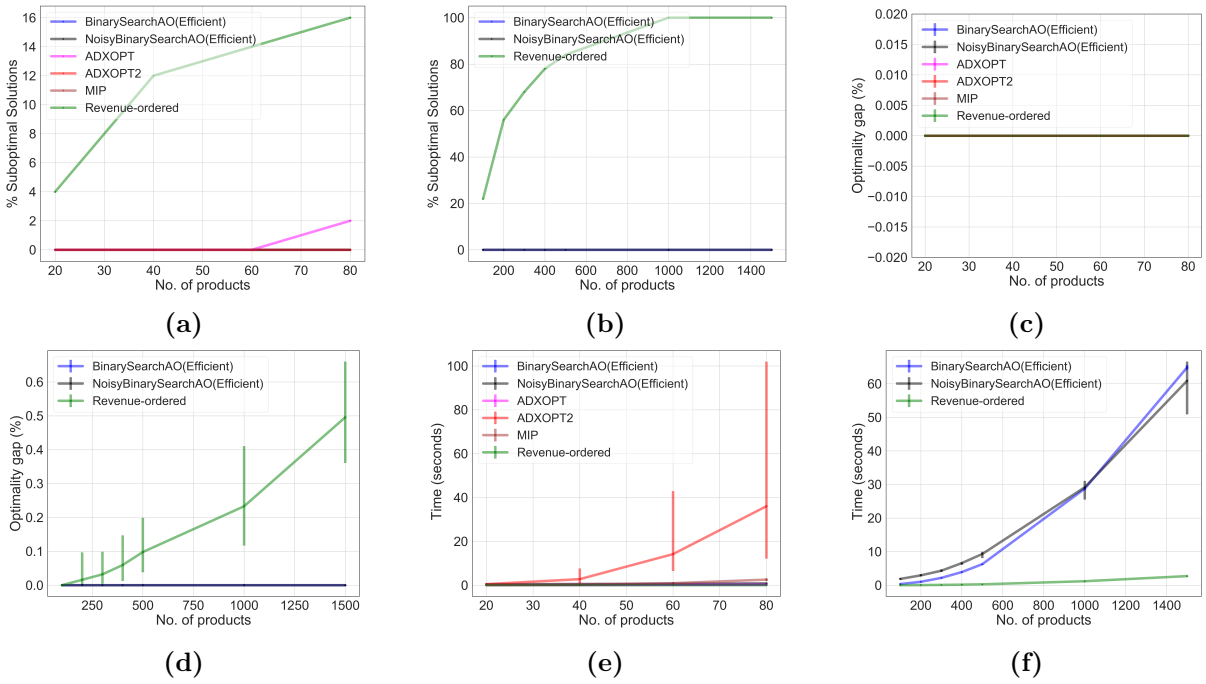


Figure 4.3: Optimality gap and run-time analysis for the unconstrained optimization problem on the Ta Feng dataset. *Fraction of suboptimal solutions:* In ((a)), BinarySearchAO(Efficient), NoisyBinarySearchAO(Efficient), Adxopt2 and Adxopt return no suboptimal solution when number of products $n \leq 60$. In ((b)), revenue-ordered heuristic seems to trail in terms of obtaining optimal solutions. *Optimality gap:* In ((c)), all the algorithms have the median optimality gap as 0. In ((d)), BinarySearchAO(Efficient) and NoisyBinarySearchAO(Efficient) curves overlap and very close to 0. Also, the gaps for revenue-ordered are small in the context of this dataset. *Run-times:* In ((e)) Adxopt, MIP, BinarySearchAO and revenue-ordered curves are close to zero, but they increase a lot for larger sizes. In ((f)) we see that revenue-ordered heuristic is much faster as expected.

using CPLEX version 12.9.0 while the others are based on Python 3.7.4/Numpy 1.17, with some custom C++ code for the QUBO heuristics. We rely on two measures: (a) the fraction of times the algorithm generates a suboptimal solution, and (b) median optimality gap over all Monte Carlo runs. Utilizing structural properties improves solution quality and leads to faster computations because number of products to optimize over in subsequent Compare-Steps keeps decreasing. Thus, we use BinarySearchAO(Efficient) instead of BinarySearchAO for this experiment.

Figures 4.3a, 4.3b and 4.3c, 4.3d show the fraction of times suboptimal solutions were returned and the optimality gap of various algorithms respectively. From these plots, we note

that NoisyBinarySearchAO(Efficient) has good performance on all instance sizes. It converges to the optimal solution for more than 90% of the instances for all regimes of product sizes. We also note that the other heuristic approaches - the revenue-ordered heuristic, Adxopt and Adxopt2 - also output close to optimal solutions, even though the revenue-ordered heuristic fails to return an optimal solution most of the time.

Next, we compare the run-times of these algorithms in Figures 4.3e and 4.3f. We observe that the run-time of the two greedy algorithms (Adxopt and Adxopt2) and also the MIP solver doesn't scale well as the number of products increases. Moreover, the greedy algorithms have a large variance in run-times. Consequently, we limit their evaluation to the small regime (where the number of products is less than 100). Among the binary search based approaches, NoisyBinarySearchAO(Efficient) becomes faster in comparison to exact binary search as number of products increase. Also, as expected, the revenue-ordered heuristic takes the least time to run. Overall, the binary search and revenue-ordered algorithms scale well as the number of products increase when compared to other approaches. Between binary search approaches and the revenue-ordered heuristic, the former have a lower optimality gap (with direct implications on expected revenue). While one could make a case about trading-off sub-optimality with speed in favor of the revenue-ordered heuristic, we illustrate later in this section how the heuristic's performance is very sensitive to the datasets. In particular, it exhibits large optimality gaps for some natural synthetically generated datasets.

3. Benchmarking algorithms for the BundleMVL-2 model (constrained setting): In this experiment, we evaluate the performance of algorithms when there is a constraint on the size of the feasible recommendation set. The Adxopt and Adxopt2 heuristics are modified so that in every greedy step, the best subset of products to add is chosen while ensuring that the capacity constraint is obeyed. Similarly, for the revenue-ordered heuristic, we choose the best among all revenue-ordered sets of size less than the given capacity. The structural properties of the optimal solution under the unconstrained setting do not hold for the capacitated setting. Thus, we do not use the efficient versions of our binary search approach, and focus our evaluation to small and medium regimes of number of products. We fix the capacity constraint to 5 and 20 for these two regimes respectively.

Further, to use heuristic QUBO solvers, we need to reformulate the optimization problem with capacity constraint to an unconstrained optimization problem as described in Equation (4.7). Empirically, we observe that the heuristic QUBO solvers don't perform well in this setting.

One reason for why this happens is the following: in the unconstrained setting, we observe that the value of θ_{ij} is 0 or close to 0 for many pairs of products. As a result, the quadratic coefficient matrix is sparse, and moreover has small values in the entries where the coefficient is non-zero. In the capacity constrained problem, we add an additional term to the objective with the scalar multiplier λ taking a large negative value. Thus, all the coefficients become non-zero. This significantly changes the sparsity of the problem instance, giving us poor quality solutions when solved using the previously shortlisted QUBO heuristics.

The performance of different algorithms is presented in Figure 4.4. As discussed, the noisy binary search approach that uses heuristics does much worse in terms of solution quality as compared to binary search using an exact QUBO solver. Also, there is no significant improvement in solution quality as compared to the revenue-ordered heuristics for this dataset. Due to the capacity constraint, the greedy algorithms ADXOPT and ADXOPT2 also run much faster (when compared to their performance in the unconstrained setting). The revenue-ordered algorithm has a constant time performance as it only needs to evaluate the first d revenue-ordered sets where d is the capacity constraint, irrespective of the total number of products. Overall, we conclude that the current shortlisted QUBO heuristics are not competitive for this constrained setting when compared to other approaches. Nonetheless, the binary search approach works well with exact computation at each comparison step. It also scales well in this product size regime. We will need newer heuristics to improve the run-time if the binary search approach needs to compete with other approaches (revenue-ordered, ADXOPT or ADXOPT2) in larger product size regimes.

4. Assessing optimality of the revenue-ordered heuristic for the BundleMVL-2 model: In both the settings above, i.e., the unconstrained and the constrained optimization over the Ta Feng, as well as with the UCI dataset in the appendix, we observe that the revenue-ordered heuristic performs quite well both in terms of run-time and optimality gap. In both these datasets, the proportion of non-zero $\theta_{i,j}$s is $\sim 2\%$, which explains the efficacy of the heuristic based on our analysis (see Thm. 4.4.5). Motivated by this observation, we further test the performance of this heuristic on additional synthetically generated datasets, and observe that the optimality gap of this heuristic is not always as low as in the two real world datasets we considered before. One such synthetic data generation process for the BundleMVL-2 model is as follows: we divide the products into two groups of equal size - high priced products and low priced products. Users are very unlikely to buy two high priced products. To capture this, $\theta_{i,j}$ parameters are sampled from Beta$(1, 10)$

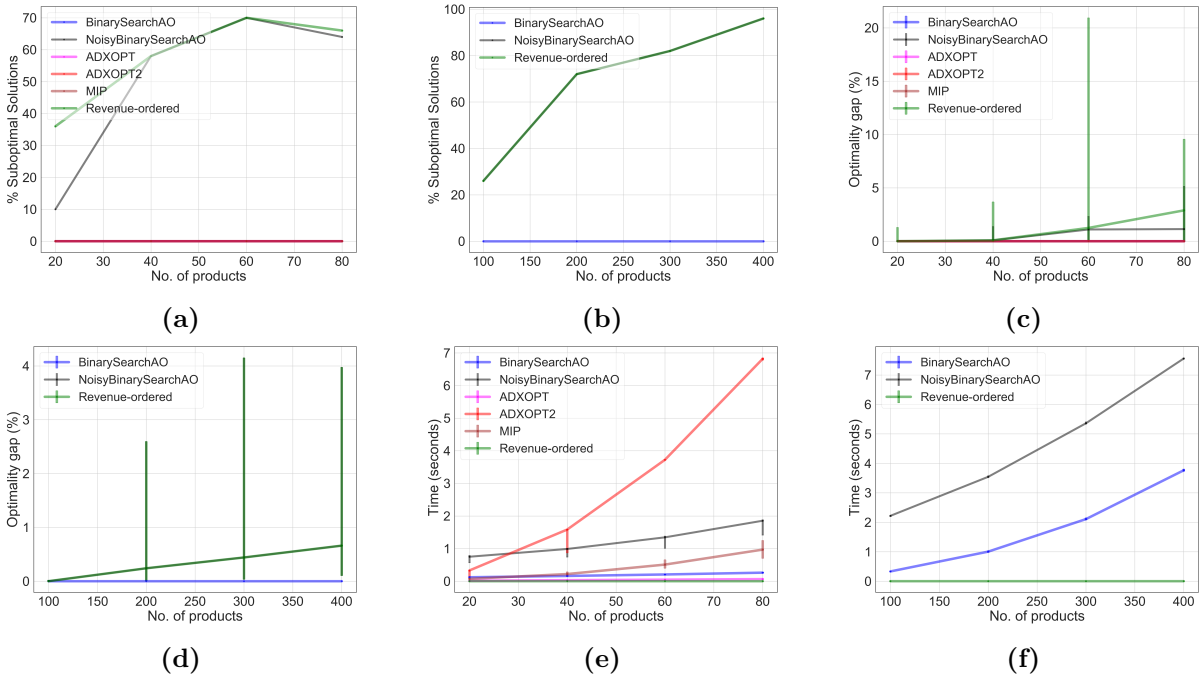


Figure 4.4: Optimality gap and run-time analysis for the optimization problem with capacity constraints on the Ta Feng dataset. ((a)): BinarySearchAO, MIP and Adxopt gets optimal solution in most runs, i.e the plots overlaps near zero value. ((b)) and ((d)): NoisyBinarySearchAO uses the revenue-ordered solution as a lower bound and generates no gain on top of it. The curves overlap completely. ((c)): Adxopt, Adxopt2, MIP and BinarySearchAO plots overlap as the median optimality gap is zero in the small product size regime. Revenue-ordered's gap improves from ((c)) to ((d)) due to an increase in the capacity constraint.

distribution when products i and j both belong to the high priced group, and the rest of the $\theta_{i,j}$'s are sampled from Beta$(10, 1)$ distribution. Additionally, the prices (r_i's) are sampled from Beta$(2, 10)$ and the θ_i parameters are sampled from Beta$(1, 1)$. Figure 4.2e shows the optimality gap of revenue-ordered and NoisyBinarySearchAO algorithms on BundleMVL-2 instances of this type, supporting our hypothesis. We observe that the revenue-ordered heuristic has higher optimality gaps as compared to those observed in the experiments with Ta Feng and UCI datasets. This sensitivity to datasets may tip the sub-optimality speed trade-off balance back in favor of our proposed binary search based approaches.

4.6. Discussion

While the focus of the paper has been on proposing a parsimonious model for multi-purchase behavior in retail and other online settings followed by algorithmic solutions for maximizing revenues, there are a few key managerial insights that become apparent. Firstly, the investigations in this paper support the importance of modeling richer consumer behavior models going beyond what has been done previously. Although operational decision based on these richer models may pose challenges, we have shown fairly extensively that scalable near real-time computation of revenue maximizing recommendation sets is entirely feasible, even if the problem is NP-complete in theory. Not only is the computation fast, but the expected revenue gains to be had with richer models make them well worth the effort. Second, practitioners should not be afraid of the increase in complexity with these models. For instance, the proposed models in this work are parametric and interpretable, and only depend quadratically on the number of products, a feature common with many single-choice models such as the Markov chain choice model, the paired combinatorial logit model and others.

Third, managers and practitioners can easily trade-off model complexity with increase in optimization times with the models and algorithms proposed here. The richer the model, the better it represents consumer purchase behavior, while at the same time increasing the operational aspects (such as run-times). In fact, the iterative nature of the proposed algorithm allows one to control this trade-off in a fine grained manner: they can choose less number of iteration steps to solve for candidate recommendation sets, which may not be optimal but useful to improve the user experience and revenues compared to the alternative. Further, BundleMVL-2 model from the BundleMVL-K family seems to be rich enough to provide tangible gains when

compared to single-choice models.

Fourth, the search for scalable solutions should be a high priority given the level of personalization that consumers expect from these platforms today. Techniques ranging from segmenting products into clusters, or inducing sparsity in the BundleMVL-2 by thresholding smaller β_{ij}s to zero, can help achieve this goal, in addition to the use of structural properties and heuristics that we demonstrated here. Finally, a key insight useful for decision makers and practitioners that emerged from the experiments conducted in this paper is the remarkable effectiveness of provably suboptimal heuristics on real world data. While theory only gives limited insights on the optimality performance of heuristics such as the revenue-ordered heuristic or ADXOPTL, we empirically observe that they were fairly competitive to the best performing binary search based approach that we proposed for two real world datasets. This implies that one should rely on real world data related to the specific problem at hand while making algorithmic as well as modeling choices. The interpretability and simplicity advantages of these heuristics are an added bonus in this regard.

4.7. Conclusion

In this work, we evaluated the effectiveness of multi-purchase choice behavior captured via the proposed BundleMVL-K family on improving revenues, and compared it to other state-of-the-art models. Significant gains, of the order of $6-8\%$ relative expected revenue improvement, is observed from incorporating multiple purchases (with 1500 products, as compared to the MNL baselines). This linking between revenue/sales gains and multi-purchase behavior models is a key contribution of our work, as almost all prior work in this area was focused on modeling/estimation compared to revenue maximization (we also showed a relative 17% improvement in likelihood fits across 8 datasets). Although the gains are significant, the optimization problems are much harder than those for single-choice models, so we designed scalable algorithms that allow a practitioner to realize these gains in demanding applications such as e-commerce platforms.

Chapter 5

Near Optimal A-B Testing

5.1. Introduction

The prototypical example of an 'A-B test' is the design of a clinical trial where one must judge the efficacy of a treatment or drug relative to some control. In a different realm, A-B testing today plays an increasingly pivotal role in e-commerce, ranging from the optimization of content and graphics for online advertising, to the design of optimal layouts and product assortments for webpages. E-commerce properties will even use A-B testing as a means of finding the best third party vendor for a specific service on their website (such as, say, recommendations or enterprise search).

A natural approach to A-B testing is to independently, and with equal probability, assign each subject to either the treatment or control groups. Following such a randomized allocation, the benefit of the treatment relative to the control can be estimated from the outcomes of subjects in the two groups. The notion of a subject here can range from a patient in the clinical trial setting to a web-surfer or impression in the e-commerce setting. Similarly, the notion of a treatment can vary from an actual medical treatment in the clinical trial setting to the decision to show a specific ad in the e-commerce setting. While randomized allocation is simple and can easily be shown to yield unbiased estimates of the treatment effect under a minimal set of assumptions, the *efficiency* of this procedure (or, the sample size needed to get a statistically significant estimate of the treatment effect) can prove onerous in practice. To see, why consider the following challenges:

1. Limited Sample Size: In the clinical trial setting, the number of subjects is limited for several reasons. As an example, the cost of managing a single subject through a clinical trial

is tens of thousands of dollars (see, e.g., Steensma and Kantarjian 2014). In the e-commerce setting, one may need to conduct many thousands of A-B tests in an ongoing fashion. As an example, consider an advertising firm that uses A-B testing on live impressions (i.e., web-surfers) to mechanically decide the appropriate messaging, text size, font, color etc. for the creative content it generates for an online advertising campaign. In this domain, a reduction in the sample size needed to learn can, due to scale, result in dramatic, continual cost savings.

2. Confounding Effects: Running counter to the need for quick inference, the impact of a particular treatment (or design decision) may be marred by a potentially large number of covariates. The presence of these covariates makes the inference of the treatment effect more challenging, since the difference in outcome of the treatment and control groups might be due to a lack of 'balance' in the covariates in the two groups. While the law of large numbers assures us that a large enough sample size will 'wash out' the impact of this imbalance of covariates, the requisite sample size may grow exceedingly large when the number of covariates is large and/or the treatment effect is small.

3. 'Small' Treatment Effects: Similar to the covariate imbalance issue above, the incremental impact of the treatment under study may be relatively 'small'. This creates a challenge in the measurement of small treatment effects, which, despite their magnitude, many nevertheless be important in settings where the selected treatments will be applied on a sufficiently large scale. More precisely, if one imagined a model where the outcome is additively impacted by the treatment and exogenous noise, we expect the sample size required to discern the treatment from noise to grow quadratically with the ratio of the standard deviation of the exogenous noise to the treatment effect. To (heuristically) see why, observe that if S_n is the sum of n independent, zero mean random variables, each with standard deviation σ, $\theta > 0$ is some constant, and $\Phi(\cdot)$ is the cumulative distribution of the standard normal, then by the central limit theorem, we expect

$$\mathbb{P}\left(\left|\frac{S_n}{n}\right| \geq \theta\right) \sim 2\Phi\left(\frac{\theta\sqrt{n}}{\sigma}\right).$$

This suggests that, in order to differentiate a treatment effect with magnitude θ from exogenous noise with standard deviation σ, we need on the order of σ^2/θ^2 samples.

4. Operational Constraints: As already alluded to, A-B tests can be expensive, either because of an explicit cost related to managing test subjects or the implicit risk of testing a sub-optimal treatment. These issues clearly impact the choice of sample size and frequently imply a budget on the number of subjects allocated to the alternative treatment whose efficacy we seek to measure. It is also not unusual to dynamically 'stop' a trial based on ones confidence in the outcome. In clinical trials, one cares about 'selection bias' in addition to efficiency; measures such as selection bias speak to concerns of robustness (to modeling errors or manipulation), or even fairness. Taken together, these operational constraints further complicate an already challenging problem.

Addressing theses challenges motivates considering the careful design of such A-B tests. In particular, given a collection of subjects, some of whom must be chosen for treatment, and others assigned to a control, we would like an assignment that 'balances' the distribution of covariates across the two groups. This in turn could conceptually yield an efficient estimate of the treatment effect, the primary concern alluded to above.

Given the broad applicability of an efficient A-B test, it is perhaps not surprising that a large body of literature within the statistical theory of the design of experiments has considered this very problem, starting with the nearly century old work of Fisher (1935). While we defer a review of this substantial literature to Section 5.1.2, a very popular approach to dealing with the problem of achieving covariate balance is the use of 'stratification'. In this approach, the subjects are divided into a number of groups based on the covariates. In other words, the covariate space is divided into a number of regions and subjects whose covariates lie in a certain region are grouped together. Further, each of the groups is randomly split to be allocated to the treatment or the control. Unfortunately, stratification does not scale gracefully with the number of covariates since the number of groups required in stratification will grow exponentially with the dimension.[1] Another natural idea would be to 'match' subjects with similar covariates, followed by assigning one member of a match to the treatment and the other to the control. Such a design would try to mimic an idealistic scenario in which, for n subjects under the experiment, we have $n/2$ pairs of 'twins'. If the matched subjects are indeed close to each other in the space of covariates, we would have that the distribution of covariates in the treatment and control is close to each other, which would cancel out the effect of these covariates. While this latter

[1] Rule C in Table 1 of Atkinson (2002) illustrates that with as few as 10 covariates, methods based on stratification are hardly better than randomization.

approach does allow us to consider a large number of covariates, the literature only appears to present heuristics motivated by these ideas.

To add a further challenge beyond those already discussed, an additional (and very important) requirement apparent from the applications above is that the process of allocating subjects (or impressions) to a particular treatment (or creative) must be made *sequentially, in an online or dynamic fashion.* Again, there is a literature on dynamic allocation starting with seminal work by Efron (1971) on 'biased coin designs' (BCDs). While a BCD seeks to balance the number of subjects in the treatment and control groups, there is by now a robust literature on so-called covariate adaptive BCDs. These schemes extend Efron's original proposal so that one cares about balance in not just the number of subjects across the two groups but also seeks balance in the covariate distribution. Viewed from the perspective of dynamic optimization, all of these heuristics can be seen as myopic schemes that in making an allocation at a given point in time fail to hedge against the future stream of arriving subjects. In fact, the literature surprisingly does not consider the design of an 'optimal' online allocation of subjects to treatments — or online A-B testing in our parlance — as a principled dynamic optimization problem where dynamic programming techniques for optimal sequential decision-making can be applied.

The present paper casts the problem of computing an efficient estimate of the treatment effect in an A-B test as a dynamic optimization problem. Despite this being a high-dimensional control problem, we show that one can efficiently compute near-optimal solutions to this problem when covariates are elliptically distributed. We show that our approach yields Pareto improvements over state of the art alternatives covariate adaptive BCD approaches. As a secondary contribution, we also show that that the important 'offline' variant of the problem also admits an efficient optimal algorithm and tightly characterize the value of optimization in that setting.

5.1.1. This Paper

Our approach, in a nutshell, is to formulate online A-B testing as a (computationally challenging) dynamic optimization problem and develop approximation and exact algorithms for the same. In particular, the present paper considers the setting where a subject's response is linear in the treatment and covariates; as we discuss later, this is a canonical model and is widely encountered in the literature on experiment design. We consider the problem of maximizing the precision of our estimate of the treatment effect by optimally allocating subjects to either the treatment or control group. We formulate this problem as a dynamic optimization problem and make the

following contributions:

1. Offline Allocation: In the offline setting, i.e., where the allocation can be made after observing all subjects, we show that the problem can be solved efficiently by using as a subroutine a generalization of the MAX-CUT SDP relaxation of Goemans and Williamson (1995). While not our main result, this result shows that the problem of *offline* A-B testing (which is still valuable in some traditional applications) can surprisingly be solved efficiently. We also characterize the value of optimized allocations relative to randomization in this setting and show that this value grows large as the number of covariates grows.

2. Sequential Allocation: In the online setting — which is the algorithmic focal point of our work — our optimization problem is, not surprisingly, a high dimensional dynamic optimization problem with dimension that grows like the number of covariates. *We show how to break the curse of dimensionality here.* In particular, we show that the state space of this dynamic optimization problem collapses if covariates come from an elliptical family of distributions (a family that includes, for example, the multivariate Gaussian). This yields an *efficient* algorithm that is provably optimal in the elliptical distribution setting and that can nonetheless be employed when covariates are not from an elliptical family.

3. A General Framework: We show that our dynamic optimization formulation permits the consideration of criteria beyond just the variance of the treatment effect. Specifically, we extend our formulation to a framework that can accommodate the simultaneous minimization of selection bias; the minimization of general separable cost functions of the allocation; endogenous (optimal) stopping criteria (as opposed to a-priori fixed sample sizes); and budgets on the sample size for a given treatment, to name just a few applications of the framework.

4. Experimental Comparisons: We compare our approach to sequential allocation with a host of so-called covariate adaptive BCD approaches, several of which are considered state-of-the-art. It is typical to measure the performance of such approaches not just in terms of efficiency, but also with respect to the so-called selection bias they induce. Here we show that our approach yields a *Pareto improvement* over these alternatives. In addition to synthetic data, we run our experiment on real user impression data from Yahoo.com. We show similar Pareto gains despite the fact that the covariates in the real data are categorical.

Thus, our main contribution is providing an algorithm for the challenging problem of sequential A-B testing that can be shown to be near-optimal when covariates are drawn from an elliptical family. The algorithm is applicable to a canonical family of treatment models and also applies to the simultaneous optimization of several criteria. Given the vast extant literature on this problem, and the fact that it is nominally high-dimensional, it is a pleasant surprise that such an algorithm exists.

5.1.2. Related Literature

The theory of optimal experiment design (which, in a sense, subsumes the problems we consider here) starts with the seminal work of Fisher (1935). Important textbook expositions of this mature topic include that of Pukelsheim (2006) and Cook et al. (1979), the latter of which discusses the notion of covariate matching as it applies to practice. While not our primary focus, the 'offline' problem we discuss in this paper is of practical relevance in the social sciences; see Raudenbush et al. (2007), for an application and heuristics. Kallus (2013) studies an approach to this problem based on linear mixed integer optimization with an application to clinical trials. In a follow-up paper, Bertsimas et al. (2015) presents a robust optimization framework for the offline problem with an emphasis on allocations of treatments that are robust to the specific form of the model of each subject's response as a function of the treatments and subject covariates (we merely consider linear functions here). The value of optimization has also recently received attention from the economics community; Kasy (2013) discusses several optimization formations that complement those proposed by Bertsimas et al. (2015). Unlike Kallus (2013), Bertsimas et al. (2015) however, Kasy (2013) offers no algorithmic approach to solve the problems he proposes (and unfortunately, his problem formulations appear largely intractable). In contrast, we focus on a class of models where the treatment effect is linear in the observed covariates and offer efficient approximation algorithms for the same. Our formulation is closely related to the case of squared loss with a non-informative prior in the verbiage of Kasy (2013). Our offline problem may be viewed as a special case of the problem of D_a-optimal experiment design and fortuitously coincides with an optimality criterion that already enjoys wide acceptance. By virtue of their computational efficiency, our techniques can be brought to bear in settings where the size of the problem can be very large rendering brute-force techniques for optimization (such as those suggested by Kasy (2013)) infeasible.

The problem that is of greatest algorithmic interest to us is the 'online' allocation problem,

where treatments must be assigned to subjects as they arrive. With regard to this sequential problem, Efron (1971) proposed an allocation strategy, referred to as a 'biased coin design' (BCD), that sought to 'balance' the number of subjects in each trial while minimizing certain types of selection bias. Now whereas Efron's BCD seeks only to balance the number of subjects between test and control groups, there is by now a robust literature on so-called covariate adaptive BCDs (CA-BCDs). Such schemes seek balance not just in the number of subjects but also in the covariate distribution between groups. Perhaps the most widely used CA-BCD is the procedure proposed by Pocock and Simon (1975) wherein the authors recommend a 'bias' that depends on a generic cost function of the covariate imbalance between the two groups. Atkinson (1982, 1999) proposed the first CA-BCD whose design is rooted in theory, specifically to the notion of D_a optimality in experiment design; of course this approach comes at the cost of assuming a treatment effect model. A number of model-based CA-BCD proposals have followed, including Smith's rule (Smith 1984b,a); the Bayesian procedure of Ball et al. (1993); and rule ABCD, proposed by Baldi Antognini and Zagoraiou (2011), to name a few. The so-called minimization approach of Pocock and Simon (1975) (which applies to generic cost functions of covariate imbalance) has also been recently analyzed by Hu and Hu (2012), who prescribe a more refined class of cost functions that lead to asymptotic balance. Alternatives to the CA-BCD procedure have also been proposed recently: Kapelner and Krieger (2014) presents an approach to achieving covariate balance based on ideas from the theory of online matching.

Viewed from the perspective of dynamic optimization, except for the heuristic proposed by Kapelner and Krieger (2014), all of the above approaches can be regarded as *myopic* policies. Such policies only consider the immediate impact of an allocation decision, and do not consider the impact on future decisions. In general, myopic policies will not be optimal. It is worth noting that for all of the aforementioned procedures, the theoretical analysis available, if any, is always in a limiting regime where sample size grows large keeping the number of covariates fixed. Little is understood in finite samples. More generally, Rosenberger and Sverdlov (2008) note that "very little is known about the theoretical properties of covariate-adaptive designs". In contrast, we see that our approach yields provably optimal allocations in finite samples for a host of optimality criteria. As we see in our experimental work, this also translates to Pareto improvements over several of the schemes described above, even on real data. It is worth noting however, that such statements of optimality require restrictions on the types of treatment models one can consider, as well as distributional assumptions on the covariates.

Related but Distinct Problems.

It is important to distinguish the experiment design problems considered here from 'bandit' problems, particularly those with side information (e.g., Woodroofe 1979, Langford and Zhang 2007) as both classes of problems frequently find application in very related applications. In theory, the experimental design setting is appropriate when an irrevocable decision of what treatment is appropriate must be made (e.g., the number of ads to display with search results), whereas the bandit setting is appropriate in a setting where the decision can be changed over time to optimize the (say) long-run average value of some objective (e.g., maximizing revenues by finding the best audience for a specific campaign). In practice, the choice of which framework to use is frequently complicated by operational considerations. For instance consider the problem of deciding between two distinct creatives in an advertising campaign. The bandit formulation is elegant and quite natural for this setting (Hauser et al. 2009, Schwartz et al. 2017). Despite this, it is common industry practice to make such decisions using frequent A-B tests[2]. From a methodological perspective, an important difference is that solution methods for bandit problems need to address an 'exploitation-exploration' trade-off between learning the best alternative and collecting rewards to optimize the objective, while there is no such trade-off in our experimental design setting.

Other problems in marketing science are also close in spirit to the A-B testing problem we study. Adaptive conjoint analysis seeks to learn the tastes of an individual (or a group of individuals) by asking a sequence of questions (or presenting a sequence of choices). In an effort to learn accurately with as small a number of questions, Toubia et al. (2003, 2004) propose a dynamic optimization procedure that is in the spirit of the ellipsoid method in convex optimization.

Another closely related class of problems are ranking and selection problems where the task is to pick the best of a set of alternatives with a budget on samples (for an overview, see Kim and Nelson 2006). In our lexicon, the emphasis in such problems is choosing from multiple (typically, greater than two) treatments in the *absence* of observable covariates on a sample. Interestingly, recent progress on this class of problems has also heavily employed dynamic optimization techniques (see, e.g., Chick and Gans 2009, Chick and Frazier 2012, Chick et al. 2017).

As a final note, the major emphasis in our work is on A-B testing with a fixed budget on

[2]For example, consider the following case study by one of the largest providers of commercial A-B testing infrastructure: https://blog.optimizely.com/2014/02/03/case-study-sony-ab-tests-banner-ads/.

samples. It is interesting to consider A-B tests that can be 'stopped' with continuous monitoring. Doing so can introduce a significant bias towards false discovery; Johari et al. (2017) have recently made exciting progress on this problem.

5.2. Model

In this section we describe the model. Given the model assumptions in Section 5.2.1, our problem is to maximize the precision of our estimate of the treatment effect. In Section 5.2.2 we pose the two optimization problems that are of interest. One of them is the offline problem where all subjects can be observed before making allocation decisions and the other is the sequential problem where subjects must be allocated without knowing the future arrivals. In Section 5.2.3 we present a simple upper bound on the precision of any estimate of the treatment effect given an allocation; this allows us to define the notions of efficiency and loss. Section 5.2.4 concludes with an intuitive interpretation of our optimization problems.

5.2.1. Setup

We must learn the efficacy of a treatment by observing its effect on n subjects. The kth subject is assigned a treatment $x_k \in \{\pm 1\}$. The kth subject is associated with a covariate vector (i.e., side information or context) $Z_k \in \mathbb{R}^p$. We assume that impact of the treatment on the kth subject is given by:

$$y_k = x_k\theta + Z_k^\top \kappa + \epsilon_k.$$

This assumes a linear dependence of the covariates and treatment decision on the outcome. The treatment effect $\theta \in \mathbb{R}$ and the weights on the covariates $\kappa \in \mathbb{R}^p$ are unknown. Our aim is to estimate θ. The $\{\epsilon_k\}$ are i.i.d. zero mean random variables with variance σ^2. The key restriction imposed by this model is that the impact of treatment is additive, an assumption that is ubiquitous in all of the related literature on the topic. Further, we assume that there is no endogeneity, i.e. the idiosyncratic noise in the model, ϵ_k, is uncorrelated with any of the covariates in Z_k.[3]

Letting $Z \in \mathbb{R}^{n\times p}$ be the matrix whose kth row is $Z_k^\top$, throughout this paper, we will assume that:

[3]The assumption of no endogeneity is required for the least square estimate of θ under a given allocation to be unbiased. It is also required for our performance analysis. In general, it appears difficult to overcome bias in the face of the risk of model mis-specification while using a covariate dependent treatment assignment scheme.

Assumption 5.2.1. The first column of Z is a vector of all ones. Further, Z is full rank and $p \leq n-1$.

The requirement that one of the covariates be a constant ensures that θ is interpreted as a treatment effect, otherwise it could be learned from the assignment of a single treatment. The crucial assumption is that $p \leq n-1$, which nonetheless allows for a large number of covariates.[4] In fact the scenario where $p \sim n$ is particularly relevant. Our problem formulation does not apply to the regime where $p > n$; indeed a formulation that is relevant to that regime is unclear to us since treatments must be assigned prior to having observed outcomes. For a particular allocation of treatments, x, let us denote by $\hat{\theta}_x$ the least squares estimator for θ.

5.2.2. Optimization Problem

We are interested in finding an experiment design with minimal variance or, equivalently, maximal precision. A standard calculation yields that the estimator $\hat{\theta}_x$ has precision

$$\mathrm{Prec}(\hat{\theta}_x) \triangleq \frac{1}{\mathrm{Var}(\hat{\theta}_x)} = \frac{x^\top P_{Z^\perp} x}{\sigma^2}, \tag{5.1}$$

where $P_{Z^\perp} \triangleq I - Z(Z^\top Z)^{-1}Z^\top$. Details are presented in the Electronic Companion to this paper.

We can now immediately state the *offline experiment design problem*:

$$\begin{aligned} \text{(P1)} \triangleq \text{maximize} \quad & x^\top P_{Z^\perp} x \\ \text{subject to} \quad & x \in \{\pm 1\}^n. \end{aligned}$$

Here, given the collection of covariates Z, we seek to find the allocation x which yields the least squares estimate with maximal precision.

In many real world applications the assignments need to be made in a sequential fashion. Subjects arrive one at a time and the assignment must be made without the knowledge of subjects in the future. We formulate this as a dynamic optimization problem. To this end we must now assume the existence of a measure on the covariate process $\{Z_k\}$. We define a filtration $\{\mathcal{F}_k\}$ by setting, for each time k, $\mathcal{F}_k$ to be the sigma algebra generated by the first k covariates $(Z_1, \ldots, Z_k)$ and the first $k-1$ allocations $(x_1, \ldots, x_{k-1})$. The *online experiment*

[4] We will informally refer to p as the number of covariates even though, strictly speaking, it is the dimension of the linear model and could include second order terms, interaction terms between covariates, etc.

design problem is then given by:

$$\begin{aligned} \text{(P2)} \triangleq \text{maximize} \quad & \mathsf{E}\left[x^\top P_{Z^\perp} x\right] \\ \text{subject to} \quad & x \in \{\pm 1\}^n, \\ & x_k \text{ is } \mathcal{F}_k\text{-measurable}, \quad \forall\ 1 \le k \le n, \end{aligned}$$

where the expectation is over the distribution of the covariate process. Here, the objective is to maximized the expected *ex post* precision.[5]

5.2.3. Upper Bound, Efficiency, and Loss

The following upper bound on the precision of any unbiased estimator that is a straightforward consequence of the Cramér-Rao bound:

Proposition 5.2.2. If $\epsilon \sim N(0, \sigma^2 I)$, then for any covariate matrix Z and any unbiased estimator $(\hat{\theta}, \hat{\kappa})$, including non-least squares estimators, we have:

$$\mathrm{Prec}(\hat{\theta}_x) \le \frac{n}{\sigma^2},$$

an upper bound on the optimal value of both problems (P1) and (P2). For non-Gaussian noise ϵ, this upper bound still holds for all least squares estimators.

This proposition, whose proof is provided for completeness in the Electronic Companion to this paper, shows that the precision of the optimal estimator[6] is $O(n)$. Consider the case when subjects are identical, i.e., $p = 1$ and $Z_k = 1$ for all k. It is easy to note that, in this case assuming n is even, the optimal design allocates half of the subjects to either treatment. Further, the precision of such a design is n/σ^2, the optimal achievable precision. For $p > 1$ this precision is less than this value. Thus the presence of covariates only makes the inference challenging.

Motivated by Proposition 5.2.2, we define *efficiency* as the the precision of an estimator normalized by the Cramér-Rao upper bound, i.e.,

$$\mathrm{Eff}(\hat{\theta}_x) \triangleq \frac{\mathrm{Prec}(\hat{\theta}_x)}{n/\sigma^2} \le 1,$$

[5]Note that, in the online case, because of Jensen's inequality, maximizing precision and minimizing variance are no longer equivalent objectives.

[6]In what follows, given a function $f(\cdot)$ and a positive function $g(\cdot)$, as $n \to \infty$ we say $f(n) = O(g(n))$ if $\limsup_{n\to\infty} |f(n)|/g(n) < \infty$, we say $f(n) = o(g(n))$ if $\lim_{n\to\infty} |f(n)|/g(n) = 0$, we say $f(n) = \Omega(g(n))$ if $\limsup_{n\to\infty} |f(n)/g(n)| > 0$, and finally we say $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

Loss is defined as the sub-optimality of an estimator relative to the upper bound measured additively in sample units:

$$\text{Loss}(\hat{\theta}_x) \triangleq n - \sigma^2 \text{Prec}(\hat{\theta}_x) \geq 0,$$

so that

$$\text{Prec}(\hat{\theta}_x) = \frac{n - \text{Loss}(\hat{\theta}_x)}{\sigma^2}.$$

We consequently see that loss can intuitively be thought of as "the effective number of subjects on whom information is lost due to the imbalance of the design" (Atkinson 2014).

5.2.4. Problem Interpretation

Before moving on to algorithm design, we pause to interpret the offline and online problems presented above. First we begin with an intuitive interpretation of the objective. Define the imbalance vector in covariate values between the test and control groups, $\overline{\Delta}_n \in \mathbb{R}^p$, according to $\overline{\Delta}_n \triangleq \sum_{k=1}^n x_k Z_k = Z^\top x$. Notice that the empirical second moment matrix for the covariates is given by $\Gamma_n \triangleq Z^\top Z/n$. Then, it is easy to see that the objective of the offline problem (P1) reduces to

$$x^\top P_{Z^\perp} x = x^\top \left(I - Z(Z^\top Z)^{-1} Z^\top \right) x = n \left(1 - \overline{\Delta}_n^\top \Gamma_n^{-1} \overline{\Delta}_n \right).$$

Therefore, the offline problem (P1) is equivalent to minimizing the square of the weighted euclidean norm of $\overline{\Delta}_n$,

$$\left\| \overline{\Delta}_n \right\|_{\Gamma_n^{-1}}^2 \triangleq \overline{\Delta}_n^\top \Gamma_n^{-1} \overline{\Delta}_n,$$

while (P2) seeks to minimize the expected value of this quantity where the expectation is over the covariate process and our allocations. Put simply, both problems seek to minimize the aggregate imbalance of covariates between the treatment and control groups, measured according to this norm.

As a final point, we note that the measure of 'imbalance' minimized in problems (P1) and (P2) was derived assuming a least squares estimator, and it is worth noting that this choice is not arbitrary. Specifically, note that the Cramér-Rao bound dictates that, provided x and Z are independent of ϵ, and further if ϵ is normally distributed, then for *any* unbiased estimator of the treatment effect $\tilde{\theta}_x$, we have that

$$\text{Eff}(\tilde{\theta}_x) \leq \text{Eff}(\hat{\theta}_x)$$

where the right hand side quantity is the efficiency of the least square estimator. Now both

problems (P1) and (P2) seek to find an allocation x to maximize the latter quantity, or its expected value, respectively. Consequently, both problems may be interpreted as seeking an allocation of samples to the test and control group with a view to maximizing the efficiency of our estimate of the treatment effect among *all* unbiased estimators of the treatment effect.

5.3. The Offine Optimization Problem

In this section, we consider the offline optimization problem (P1). We show that this combinatorial problem permits a tractable, constant factor approximation using an SDP-based randomized rounding algorithm. Moreover, in this setting, we can analyze the effect optimization has on the precision of the estimator of the treatment effect, as compared to randomization. To this end, we first obtain the mean precision of the randomized design. Surprisingly, precision is a simple function of n and p and does not depend on the data matrix Z. We show that when $p \sim n$, the randomization is rather inefficient and the precision is $O(1)$. This can be contrasted with the upper bound on precision given by Proposition 5.2.2 which is $\Omega(n)$. To conclude the section, we analyze the performance of the optimal allocation assuming a distribution on Z. We show that for any p, the precision of optimal allocation is $\Omega(n)$. Thus concluding that when $p \sim n$, randomization can be arbitrarily bad as compared to the optimal design.

5.3.1. Approximation Algorithm for (P1)

First, we observe that there is a tractable approximation algorithm to solve the combinatorial optimization problem (P1). In particular, consider the semidefinite program (SDP) over symmetric positive semidefinite matrices $Y \in \mathbb{R}^{n \times n}$ given by[7]

$$\begin{aligned} \text{(P1-SDP)} \triangleq \quad & \text{maximize} && \operatorname{tr}\left(P_{Z^\perp} Y\right) \\ & \text{subject to} && Y_{kk} = 1, \qquad \forall\ 1 \le k \le n, \\ & && Y \succeq 0, \\ & && Y \in \mathbb{R}^{n \times n}. \end{aligned}$$

It is straight forward to see that (P1-SDP) is a relaxation of (P1) in the sense that it achieves higher objective value: given an optimal solution $\hat{x} \in \{\pm 1\}^n$ for (P1), define the symmetric positive definite matrix $\hat{Y} \triangleq \hat{x}\hat{x}^\top \in \mathbb{R}^{n \times n}$. Then, clearly $\hat{Y}$ satisfies the constraints of (P1-SDP).

[7] Here, $Y \succeq 0$ denotes that Y is a symmetric and positive semidefinite matrix.

Also, $\text{tr}\left(P_{Z^\perp}\hat{Y}\right) = \hat{x}^\top P_{Z^\perp}\hat{x}$, so the objective values for (P1) and (P1-SDP) coincide. Therefore, the optimal objective value of (P1-SDP) must be larger than that of (P1). Moreover, because it is an SDP, (P1-SDP) can be efficiently solved in polynomial time.

Based upon prior work on the MAX-CUT problem (Goemans and Williamson 1995), the following result, due to Nesterov (1997), establishes that (P1-SDP) can be used as the basis of a randomized algorithm to solve (P1) with a constant factor guarantee with respect to the optimal design. The corresponding (randomized) allocation procedure is described in Algorithm 5.

Algorithm 5 Randomized allocation algorithm based on (P1-SDP).

Set $Y^* \succeq 0$ to be an optimal solution of the program (P1-SDP) given the data matrix Z.
Set the matrix $V \in \mathbb{R}^{n\times n}$ with columns $v_1, \ldots, v_n \in \mathbb{R}^n$ so that the matrix decomposition $Y^* = V^\top V$ holds.
Let $u \in \mathbb{R}^n$ be a vector chosen at random uniformly over the unit sphere.
for $k \leftarrow 1, n$ **do**

$$\tilde{x}_k \leftarrow \begin{cases} +1 & \text{if } u^\top v_k \geq 0, \\ -1 & \text{if } u^\top v_k < 0. \end{cases}$$

return $\tilde{x}$

Theorem 5.3.1. Given a data matrix $Z \in \mathbb{R}^{n\times p}$, set the allocation $\tilde{x} \in \mathbb{R}^n$ according to Algorithm 5. Then,

$$\mathsf{E}_u\left[\tilde{x}^\top P_{Z^\perp}\tilde{x}\right] \geq \frac{2}{\pi} \max_{x\in\{\pm 1\}^n} x^\top P_{Z^\perp} x,$$

where the expectation is taken over the choice of random vector u in Algorithm 5. In order words, the expected value achieved by the vector $\tilde{x}$ in the offline experiment design problem (P1) is within a constant factor $2/\pi$ of the best possible.

Proof. This theorem is a direct consequence of Theorem 3.4.2 of Ben-Tal and Nemirovski (2001). That result states that any quadratic integer optimization problem with objective $x^\top Q x$, such that $x \in \{\pm 1\}^n$, can be approximated within a relative error of $\pi/2$ using the prescribed algorithm, provided Q is positive semidefinite. Since $P_{Z^\perp}$ is positive semidefinite (indeed, it is a projection matrix), the result follows. ■

5.3.2. Optimal Allocations vs. Randomized Allocations

Randomization is the most popular technique used for A-B testing. In what follows, we will compare the performance of randomization to what can be achieved by the optimal offline allocation of (P1).

In its most basic variation, simple randomization partitions the population into two equally sized groups, each assigned a different treatment, where the partition is chosen uniformly at random over all such partitions (for simplicity, we will assume that the population is of even size). Denote by $X_{\text{rand}} \in \{\pm 1\}^n$ the random allocation generated by simple randomization, and denote by $\hat{\theta}_{X_{\text{rand}}}$ the resulting unbiased least squares estimator for θ.

Theorem 5.3.2. If n is even, given a covariate matrix Z, define the expected precision and loss of simple randomization

$$\text{Prec}_{\text{rand}} \triangleq \mathsf{E}_{X_{\text{rand}}}\left[\text{Prec}\left(\hat{\theta}_{X_{\text{rand}}}\right)\right], \qquad \text{Loss}_{\text{rand}} \triangleq \mathsf{E}_{X_{\text{rand}}}\left[\text{Loss}\left(\hat{\theta}_{X_{\text{rand}}}\right)\right],$$

where the expectations are taken over the random allocation X_{rand}. Then,

$$\text{Prec}_{\text{rand}} = \frac{n}{\sigma^2}\left(1 - \frac{p-1}{n-1}\right), \quad \text{Loss}_{\text{rand}} = \frac{n}{n-1}(p-1).$$

The proof relies on simple probabilistic arguments and is presented in the Electronic Companion to this paper. Surprisingly the precision and loss of the randomized allocation *does not* depend on the data matrix Z at all, as long as it is full rank and has a constant column.

Comparing with the upper bound of Proposition 5.2.2, we notice that in the large sample size regime where $n \to \infty$, simple randomization is asymptotically order optimal in the sense that it achieves precision that grows with order n — the maximum permitted by the upper bound of Proposition 5.2.2 — when $p \ll n$. This may not be the case when p is close to n, however. For example, if $p = n - 1$, which is the maximum value p can take under Assumption 5.2.1, then $\text{Prec}_{\text{rand}} \approx 1/\sigma^2$, which is of *constant order*. In such a case, the least squares estimator $\hat{\theta}_{X_{\text{rand}}}$ will not asymptotically converge to θ as $n \to \infty$. In general, simple randomization is asymptotically order optimal any time that $p_n = o(n)$ as $n \to \infty$.

Now we consider the performance of the least squares estimator under the optimal design that would be obtained by solving the offline experiment design problem (P1). By construction, the optimal design will clearly have precision that is at least that of the randomized procedure. We would like to understand the magnitude of the possible improvement, however, and to see if it is material. Unlike in the simple randomized case, however, the precision of the optimal design depends on the covariate matrix Z. Moreover, it is difficult to obtain a closed-form expression for this precision as a function of Z.

We can illustrate this with a simple example. Consider the case where $p = n - 1$. The

precision of the optimal design is given by

$$\sup_{x\in\{\pm 1\}^n} \frac{x^\top P_{Z^\perp} x}{\sigma^2}.$$

Since $p = n-1$, the null space of $Z^\top$ is a one dimensional subspace of $\mathbb{R}^n$. Let $y \in \mathbb{R}^n$ be a non-zero vector such that $Z^\top y = 0$ and $\|y\|_2^2 = 1$. That is, y is a unit vector in the null space of $Z^\top$. It is easy to see that $P_{Z^\perp} = yy^\top$. Thus, the precision of the optimal design is

$$\sup_{x\in\{\pm 1\}^n} \frac{x^\top yy^\top x}{\sigma^2} = \sup_{x\in\{\pm 1\}^n} \frac{\left(y^\top x\right)^2}{\sigma^2} = \frac{\|y\|_1^2}{\sigma^2}. \tag{5.2}$$

Now, consider the following two cases:

1. y has only two non-zero components given by $1/\sqrt{2}$ and $-1/\sqrt{2}$. In this case, the optimal precision is $2/\sigma^2$. Thus, in this case, randomization is within a constant factor of optimal.

2. y has entries such that $|y_i| = 1/\sqrt{n}$ and $\mathbf{1}^\top y = 0$. In this case, the precision is n/σ^2. Thus, in this case, the optimal design achieves the Cramér-Rao upper bound and the performance is a significant improvement over the randomized design.

The preceding two cases show, that depending on the covariate matrix Z (which determines the vector y in the discussion above), the performance of the optimal design may be a drastic improvement over that of the randomized design. In order to study the performance of the optimal design, we proceed by making a certain probabilistic assumption on Z. Under this assumption, we will then analyze the distribution of performance of the optimal design. For this purpose, we will assume a distribution on the covariate matrix Z as follows:

Assumption 5.3.3. Given (n, p) with $1 \le p < n$, assume that the covariate matrix $Z \in \mathbb{R}^{n\times p}$ has independent and identically distributed rows. Further, assume that for each $1 \le k \le n$, the kth row $Z_k \in \mathbb{R}^p$ satisfies $Z_{k,1} = 1$, and that the vector of all components except the first satisfies $Z_{k,2:p} \sim N(0, \Sigma)$, i.e., it is distributed according to a multivariate normal distribution with zero mean and covariance matrix $\Sigma \in \mathbb{R}^{p-1\times p-1}$.

It is easy to check that, under Assumption 5.3.3, the covariate matrix Z will satisfy the full rank condition of Assumption 5.2.1 almost surely. Consider a sequence of problems indexed by the sample size n, and where the dimension of the covariates is given by $1 \le p_n < n$. For each n, let $Z^{n,p_n} \in \mathbb{R}^{n\times p_n}$ be the data matrix satisfying Assumption 5.3.3. We have that:

Theorem 5.3.4. Suppose that Assumption 5.3.3 holds with $\Sigma = \rho^2 I$. Let x^* be an optimal design obtained by solving (P1) with covariate matrix $Z = Z^{n,p_n}$, and let $\hat{\theta}_{x^*,Z^{n,p_n}}$ be the corresponding least squares estimator of θ. Denote the precision of this estimator by

$$\mathrm{Prec}_*^{n,p_n} \triangleq \mathrm{Prec}\left(\hat{\theta}_{x^*,Z^{n,p_n}}\right).$$

Then, we have that for any $\epsilon > 0$,

$$\lim_{n\to\infty} \mathsf{P}\left(\frac{\mathrm{Prec}_*^{n,p_n}}{n} < \frac{1}{8\pi\sigma^2} - \epsilon\right) = 0,$$

where the probability is measured over the distribution of the covariates.

Theorem 5.3.4 states that, with high probability, the optimal offline optimization-based design always yields $\Omega(n)$ precision under Assumption 5.3.3. Note that this is true for all possible values of $p_n < n$ with $p_n = n-1$ being the worst case (the latter fact is established in the proof). In contrast, Theorem 5.3.2 establishes that when $p = n-1$, the precision one expects under randomized allocation is $O(1)$, so that the relative improvement from optimization for this value of p is $\Theta(n)$. In other words, if the number of covariates is comparable to the sample size, we might expect dramatic improvements over simple randomization through optimization. Moreover, while the optimal design requires solution of (P1), which may not be tractable, Theorem 5.3.1 suggests a tractable approximation which is guaranteed to achieve the same precision as the optimal design up to a constant factor.

The proof of Theorem 5.3.4 is presented in Section D.3. Here we provide a proof sketch. Let $Z^{n,p} \in \mathbb{R}^{n\times p}$ and $Z^{n,n-1} \in \mathbb{R}^{n\times n-1}$ be two covariate matrices defined on the same probability space (under the Assumption 5.3.3 with $\Sigma = \rho^2 I$) such that they are identical on the first p columns. We show that $\mathrm{Prec}_*^{n,p} \geq \mathrm{Prec}_*^{n,n-1}$. This establishes that $p = n-1$ corresponds to the worst case precision and allows us to focus on the sequence $\mathrm{Prec}_*^{n,n-1}$. We then analyze the distribution of $Z^{n,n-1}$. We show that $\mathrm{Prec}_*^{n,n-1}$ can be written down as a function of a unit vector in the null space of $(Z^{n,n-1})^\top$, say $y_n \in \mathbb{R}^n$. Further, y_n describes a random one-dimensional subspace of $\mathbb{R}^n$ that is invariant to orthonormal transformations that leave the constant vector unchanged. There is a unique distribution that has this property. We then identify the distribution and compute the precision in closed-form using this distribution. In

particular, we show that, as $n \to \infty$,

$$\frac{\text{Prec}_*^{n,n-1}}{n} \to \frac{1}{8\pi\sigma^2},$$

where the convergence is in distribution.

5.4. Sequential Problem

We now consider the online experiment design problem (P2). Here, decisions must be made sequentially. At each time k, an allocation $x_k \in \{\pm 1\}$ must be made based only on the first k covariates and any prior allocations. In other words, x_k is $\mathcal{F}_k$-measurable.

In this section we show that the optimization problem is tractable. First, we pose a surrogate problem in which the objective of (P2) is simplified. The details of this simplification are provided in Section 5.4.1. In Section 5.4.2, we show that the reduction in performance when the surrogate problem is used to device an assignment policy is negligible. Focusing on the surrogate problem, we show that the surrogate problem is a p-dimensional dynamic program in Section 5.4.3. Surprisingly, if we assume that the data generating distribution for the covariates comes from the so-called *elliptical family* then the state space collapses to two dimensions, making the dynamic program tractable. This state space collapse is presented in Section 5.4.4.

5.4.1. Formulation and Surrogate Problem

In order to formulate the sequential problem with an expected value objective, a probabilistic model for covariates is necessary. We will start by making the following assumption:

Assumption 5.4.1. Given (n, p) with $1 \leq p < n$, assume that the covariate matrix $Z \in \mathbb{R}^{n \times p}$ has independent and identically distributed rows. Further, assume that for each $1 \leq k \leq n$, the kth row $Z_k \in \mathbb{R}^p$ satisfies $Z_{k,1} = 1$, and that the vector $Z_{k,2:p} \in \mathbb{R}^{p-1}$ of all components except the first has zero mean and covariance matrix $\Sigma \in \mathbb{R}^{p-1 \times p-1}$.

Assumption 5.4.1 requires that the sequentially arriving covariates are i.i.d. with first and second moments. Assumption 5.3.3, by comparison, in addition imposes a Gaussian distribution.

Problem (P2) can be viewed as maximizing the expectation of terminal reward that is given

by

$$x^\top P_{Z^\perp} x = x^\top \left(I - Z(Z^\top Z)^{-1} Z^\top \right) x = n - \frac{1}{n} \left(\sum_{k=1}^{n} x_k Z_k \right)^\top \Gamma_n^{-1} \left(\sum_{k=1}^{n} x_k Z_k \right), \tag{5.3}$$

where the sample second moment of covariates is given by

$$\Gamma_n \triangleq \frac{1}{n} \sum_{k=1}^{n} Z_k Z_k^\top .$$

We write this matrix in block form as

$$\Gamma_n = \begin{bmatrix} 1 & M_n^\top \\ M_n & \Sigma_n \end{bmatrix},$$

where,

$$\Sigma_n \triangleq \frac{1}{n} \sum_{k=1}^{n} Z_{k,2:p} Z_{k,2:p}^\top, \quad M_n \triangleq \frac{1}{n} \sum_{k=1}^{n} Z_{k,2:p}.$$

Here, M_n and Σ_n correspond to sample estimates of the covariate mean and covariance structure, respectively.

We define, for each k, the scalar *sample count imbalance* $\delta_k \in \mathbb{R}$ and the *covariate imbalance vector* $\Delta_k \in \mathbb{R}^{p-1}$ by

$$\delta_k \triangleq \sum_{\ell=1}^{k} x_\ell, \quad \Delta_k \triangleq \sum_{\ell=1}^{k} x_\ell Z_{\ell,2:p}. \tag{5.4}$$

The terminal reward (5.3) is equal to

$$x^\top P_{Z^\perp} x = n - \frac{1}{n} \begin{bmatrix} \delta_n & \Delta_n^\top \end{bmatrix} \begin{bmatrix} 1 & M_n^\top \\ M_n & \Sigma_n \end{bmatrix}^{-1} \begin{bmatrix} \delta_n \\ \Delta_n \end{bmatrix}.$$

Problem (P2) is then equivalent to

$$\begin{aligned} \text{(P3)} \ \triangleq \ \text{minimize} \quad & \mathsf{E}\left[\begin{bmatrix} \delta_n & \Delta_n^\top \end{bmatrix} \begin{bmatrix} 1 & M_n^\top \\ M_n & \Sigma_n \end{bmatrix}^{-1} \begin{bmatrix} \delta_n \\ \Delta_n \end{bmatrix} \right] \\ \text{subject to} \quad & x \in \{\pm 1\}^n, \\ & x_k \text{ is } \mathcal{F}_k\text{-measurable}, \quad \forall\ 1 \le k \le n. \end{aligned}$$

Observe that the objective of (P3) corresponds to n times the loss of the estimator.

As $n \to \infty$, by the strong law of large numbers (under mild additional technical assumptions),

$\Sigma_n \to \Sigma$ and $M_n \to 0$ almost surely. Motivated by this fact, in developing an efficient algorithm for (P3), our first move will be to consider a surrogate problem that replaces the sample covariance matrix Σ_n with the exact covariance matrix Σ and sets the sample mean M_n to the exact mean 0:

$$\begin{aligned} (\text{P3}') \triangleq \text{minimize} \quad & \mathsf{E}\left[\delta_n^2 + \|\Delta_n\|_{\Sigma^{-1}}^2\right] \\ \text{subject to} \quad & x \in \{\pm 1\}^n, \\ & x_k \text{ is } \mathcal{F}_k\text{-measurable}, \quad \forall\ 1 \le k \le n. \end{aligned}$$

Here, given an arbitrary covariance matrix $\hat{\Sigma} \in \mathbb{R}^{p-1 \times p-1}$, we find it convenient to introduce the norm $\|\cdot\|_{\hat{\Sigma}^{-1}}$ on $\mathbb{R}^{p-1}$ defined by $\|z\|_{\hat{\Sigma}^{-1}} \triangleq (z^\top \hat{\Sigma}^{-1} z)^{1/2}$. In the present context, this norm is typically referred to as a Mahalanobis distance.

The roles of the sample count imbalance δ_n and the covariate imbalance vector Δ_n in the surrogate problem (P3′) are intuitive: requiring δ_n to be small balances the number of assignments between the two treatments (the focus of the so-called biased-coin designs). Requiring the same of Δ_n will tend to 'balance' covariates — when Δ_n is small, the empirical moments of the covariates across the two treatments are close. As discussed in the introduction, heuristics developed in the literature on the design of optimal trials tend to be driven by precisely these two forces.

For the rest of this section we will focus on the surrogate problem. We want to first justify the use of the surrogate objective. We do this by providing an approximation guarantee in Section 5.4.2. We then turn our attention on how to solve the surrogate problem via dynamic programming in the subsequent sections.

5.4.2. Approximation Guarantee for the Surrogate Problem

First, we show that the policy obtained by solving (P3′) is near optimal. Denote by $\hat{\mu}$ the measure over the sequence x_k induced by an optimal solution for the surrogate control problem (P3′), and let μ^* denote the measure induced by an optimal policy for our original dynamic optimization problem (P3). Now, δ_n and Δ_n are random variables given an allocation policy. Given a allocation policy μ, define

$$D_\mu^{n,p} \triangleq \mathsf{E}_\mu \left[\begin{bmatrix} \delta_n & \Delta_n^\top \end{bmatrix} \Gamma_n^{-1} \begin{bmatrix} \delta_n \\ \Delta_n \end{bmatrix} \right]$$

to be the objective value of (P3) under the allocation policy μ with sample size n and covariate dimension p. The following result is demonstrated, without loss of generality, under the assumption that Σ is the identity (otherwise, we simply consider setting $Z_{k,2:p}$ to $\Sigma^{-1/2}Z_{k,2:p}$):

Theorem 5.4.2. Suppose that Assumption 5.3.3 holds with $\Sigma = I$ and let $\epsilon > 0$ be any positive real number. Consider a sequence of problems indexed by the sample size n, where the dimension of the covariates is given by $1 \leq p_n < n$ and $\gamma_n > 0$ are real numbers such that, for n sufficiently large, $n \geq L \max(p_n, l \log 2/\gamma_n)/\epsilon^2$. Then, as $n \to \infty$

$$D_{\hat{\mu}}^{n,p_n} \leq \left(\frac{1+\epsilon}{1-\epsilon}\right)^2 D_{\mu^*}^{n,p_n} + \gamma_n n^2 + \gamma_n n^2 p_n + O\left(\sqrt{\frac{n}{p_n - 1}}\right).$$

Here, L and l are universal constants. In particular, selecting $\gamma_n \propto 1/n^4$ yields

$$D_{\hat{\mu}}^{n,p_n} \leq \left(\frac{1+\epsilon}{1-\epsilon}\right)^2 D_{\mu^*}^{n,p_n} + O\left(\sqrt{\frac{n}{p_n - 1}}\right). \tag{5.5}$$

The result above relies on the use of non-asymptotic guarantees on the spectra of random matrices with sub-Gaussian entries and can be found in the Electronic Companion to this paper.

The preceding result bounds the objective of the problem (P3) when (P3′) is used to devise an allocation policy. However, we are interested in the objective of the problem problem (P2), which is the precision or inverse variance of the design corresponding to the policy used. In particular, denote by $\text{Prec}_{\mu}^{n,p}$ the expected precision of the estimator when allocations are made with a policy μ, for a problem with sample size n and covariate dimension p, i.e.,

$$\text{Prec}_{\mu}^{n,p} = \frac{\mathsf{E}_{\mu}\left[x^{\top} P_{Z^{\perp}} x\right]}{\sigma^2} = \frac{n - D_{\mu}^{n,p}/n}{\sigma^2}. \tag{5.6}$$

Then, we have the following:

Corollary 1. *Suppose that Assumption 5.3.3 holds with $\Sigma = I$. Consider a sequence of problems indexed by the sample size n, where the dimension of the covariates is given by $1 \leq p_n < n$, and a fixed positive real number $\epsilon > 0$ such that*

$$\epsilon > \sqrt{L \limsup_{n \to \infty} p_n/n},$$

for a universal constant L. Then, as $n \to \infty$,

$$\frac{\text{Prec}_{\hat{\mu}}^{n,p_n}}{\text{Prec}_{\mu^*}^{n,p_n}} \geq 1 - \frac{4\epsilon^3}{(L-\epsilon^2)(1-\epsilon^2)} + o(1).$$

Corollary 1 gives the multiplicative loss in the precision by using an allocation derived from the surrogate problem (P3′). The multiplicative loss depends on the ratio p/n, which is captured in the choice of ϵ. For small values of ϵ the ratio of precision obtained by solving (P3′) and (P2) approaches 1. Note that this result holds in an asymptotic regime where p and n both increase to infinity, as long as p/n remains small.

Proof of Corollary 1. Consider (5.5) in Theorem 5.4.2. This holds when

$$n \geq \frac{L \max(p_n, l \log 2/\gamma_n)}{\epsilon^2}$$

with $\gamma_n = b/n^4$ for some constant b. Equivalently,

$$n \geq \frac{L \max(p_n, 4l \log n + 2l \log b)}{\epsilon^2}.$$

For n sufficiently large, clearly the constraint that $n \geq L(4l \log n + 2l \log b)/\epsilon^2$ will be satisfied. Therefore, combined with the lower bound hypothesized for ϵ, (5.5) holds as $n \to \infty$.

Using (5.6),

$$\begin{aligned}
\text{Prec}_{\mu^*}^{n,p_n} - \text{Prec}_{\hat{\mu}}^{n,p_n} &= \frac{D_{\hat{\mu}}^{n,p_n} - D_{\mu^*}^{n,p_n}}{n\sigma^2} \\
&\leq \frac{\frac{(1+\epsilon)^2}{(1-\epsilon)^2} D_{\mu^*}^{n,p_n} - D_{\mu^*} + O\left(\sqrt{\frac{n}{p_n - 1}}\right)}{n\sigma^2} \\
&= \frac{4\epsilon D_{\mu^*}^{n,p_n}}{n\sigma^2(1-\epsilon)^2} + o(1) \\
&= \frac{4\epsilon}{(1-\epsilon)^2}\left(\frac{n}{\sigma^2} - \text{Prec}_{\mu^*}^{n,p_n}\right) + o(1).
\end{aligned} \tag{5.7}$$

The first inequality follows from Theorem 5.4.2 and the last equality from (5.6).

Let $\text{Prec}_{\text{rand}}^{n,p_n}$ denote precision of the randomized policy. Using Theorem 5.3.2 and the optimality of μ^*, we have that

$$\frac{n}{\sigma^2} - \text{Prec}_{\mu^*}^{n,p_n} \leq \frac{n}{\sigma^2} - \text{Prec}_{\text{rand}}^{n,p_n} = \frac{n}{\sigma^2}\frac{p_n - 1}{n-1} \leq \frac{n}{\sigma^2}\frac{p_n}{n} \leq \frac{\epsilon^2 n}{L\sigma^2}, \tag{5.8}$$

where the last inequality uses the fact that, by hypothesis, $p_n/n \leq \epsilon^2/L$. Substituting this into (5.7) we get that

$$\mathrm{Prec}_{\mu^*}^{n,p_n} - \mathrm{Prec}_{\hat{\mu}}^{n,p_n} \leq \frac{4\epsilon^3 n}{(1-\epsilon)^2 L\sigma^2} + o(1).$$

Now, using (5.8) we get that,

$$\mathrm{Prec}_{\mu^*}^{n,p_n} \geq \frac{n}{\sigma^2}\left(1 - \frac{\epsilon^2}{L}\right).$$

Thus, we have that,

$$\begin{aligned} 1 - \frac{\mathrm{Prec}_{\hat{\mu}}^{n,p_n}}{\mathrm{Prec}_{\mu^*}^{n,p_n}} &\leq \frac{4\epsilon n}{\mathrm{Prec}_{\mu^*}^{n,p_n}(1-\epsilon)^2 L\sigma^2} + o(1) \\ &\leq \frac{4\epsilon^3}{(L-\epsilon^2)(1-\epsilon^2)} + o(1). \end{aligned}$$

This yields the result. ■

5.4.3. Dynamic Programming Decomposition

It is not difficult to see that (P3′) is a terminal cost dynamic program with state $(\delta_{k-1}, \Delta_{k-1}) \in \mathbb{R}^p$ at each time k. The pair (δ_k, Δ_k) can be interpreted as the post-decision state of the dynamic decision problem immediately after the kth allocation. In other words, given the past arrival sequence and actions, (δ_k, Δ_k) summarizes the the impact of this 'past' on the future objective. This is formally stated in the following proposition:

Proposition 5.4.3. Suppose that Assumption 5.4.1 holds. For each $1 \leq k \leq n$, define the function $Q_k : \mathbb{R} \times \mathbb{R}^{p-1} \to \mathbb{R}$ by the Bellman equation

$$Q_k(\delta_k, \Delta_k) \triangleq \begin{cases} \delta_n^2 + \|\Delta_n\|_{\Sigma^{-1}}^2, & \text{if } k = n, \\ \mathsf{E}\left[\min_{u \in \{\pm 1\}} Q_{k+1}(\delta_k + u, \Delta_k + uZ_{k+1,2:p})\right], & \text{if } 1 \leq k < n. \end{cases} \tag{5.9}$$

Then,

1. At each time k, the optimal continuation cost for the dynamic program (P3′) is given by $Q_k(\delta_k, \Delta_k)$. In other words, this is the expected terminal cost, given then covariates observed and the allocations made up to and including time k, assuming optimal decisions are made at all future times.

2. Suppose the allocation x_k^* at each time k is made according to

$$x_k^* \in \underset{u\in\{\pm 1\}}{\arg\min}\ Q_k\left(\delta_{k-1}+u, \Delta_{k-1}+uZ_{k,2:p}\right).$$

Then, the sequence of allocations x^* is optimal for the online experiment design problem (P3′).

Proposition 5.4.3, whose proof is presented in the Electronic Companion to this paper, suggests a standard dynamic programming line of attack for the surrogate problem (P3′): optimal continuation cost functions $\{Q_k\}_{1\le k\le n}$ can be computed via backward induction, and these can then be applied to determine an optimal policy. However, the dimension of this dynamic program is given by the number of covariates p. In general, the computational effort required by this approach will be exponential in p — this is the so-called curse of dimensionality. Thus, outside of very small numbers of covariates, say, $p \le 3$, the standard dynamic programming approach is intractable. However, as we will now see, that the surrogate problem surprisingly admits an alternative, low dimensional dynamic programming representation.

5.4.4. State Space Collapse

Proposition 5.4.3 yields a dynamic programming approach for the surrogate problem (P3′) that is intractable for all but very small values of p. What is remarkable, however, is that if the covariate data is assumed to have an *elliptical distribution*, then (P3′) can be solved via a tractable *two*-dimensional dynamic program. We first present the technical definition.

Definition 5.4.4. A random variable X taking values in $\mathbb{R}^m$ has an elliptical distribution if the characteristic function $\varphi\colon \mathbb{C}^m \to \mathbb{C}$ has the form

$$\varphi(t) \triangleq \mathsf{E}\left[\exp(it^\top X)\right] = \exp(i\mu^\top t)\Psi(t^\top \Sigma t),$$

for all $t \in \mathbb{C}^m$, given some $\mu \in \mathbb{R}^m$, $\Sigma \in \mathbb{R}^{m\times m}$, and a characteristic function $\Psi\colon \mathbb{C} \to \mathbb{C}$.

Elliptical distributions, studied extensively, for example, by Cambanis et al. (1981), are a generalization of the multivariate Gaussian distribution. The name derives from the fact that if an elliptical distribution has a density, then the contours of the density are ellipsoids in $\mathbb{R}^m$ parameterized by μ and Σ. A useful standard result for us (see, e.g., Cambanis et al. 1981) is

that these distributions can be generated by independently generating the direction and the length of the deviation (in $\|\cdot\|_{\Sigma^{-1}}$-norm) from the center μ:

Proposition 5.4.5. If X has an elliptical distribution with parameters μ, Σ, and Ψ, then there exists a non-negative random variable R such that,

$$X \stackrel{d}{=} \mu + R\Sigma^{1/2}U,$$

where U is distributed uniformly on the unit sphere $\{x \in \mathbb{R}^{p-1} \mid \|x\|_2^2 = 1\}$ and U and R are independent.

Thus, any elliptical distribution can be identified with a vector $\mu \in \mathbb{R}^m$, a positive semidefinite matrix $\Sigma \in \mathbb{R}^{m\times m}$, and random variable R taking values on the non-negative real line. We denote such a distribution by $\mathrm{Ell}(\mu, \Sigma, R)$. It can be shown that if $R^2 \sim \chi_m^2$ is a chi-squared distribution with m degrees of freedom, then $\mathrm{Ell}(\mu, \Sigma, R)$ is a Gaussian distribution with mean μ and covariance Σ. Well-known distributions such as the multivariate t-distribution, Cauchy distribution, and logistic distribution also fall in the elliptical family.

We state the assumption needed for the state space collapse.

Assumption 5.4.6. Given (n, p) with $1 \le p < n$, assume that the covariate matrix $Z \in \mathbb{R}^{n\times p}$ has independent and identically distributed rows. Further, assume that for each $1 \le k \le n$, the kth row $Z_k \in \mathbb{R}^p$ satisfies $Z_{k,1} = 1$, and that the vector $Z_{k,2:p} \in \mathbb{R}^{p-1}$ of all components except the first is distributed according to $\mathrm{Ell}(0, \Sigma, R)$, where it is assumed that the random variable R has finite second moment, and further that, without loss of generality,[8] $\mathsf{E}[R^2] = p - 1$.

The following theorem shows how the p-dimensional dynamic program is reduced to a 2-dimensional one with Assumption 5.4.6.

Theorem 5.4.7. Suppose that Assumption 5.4.6 holds. For each $1 \le k \le n$, define the function $q_k \colon \mathbb{Z} \times \mathbb{R}_+ \to \mathbb{R}$ according to

$$q_k(m, \lambda) \triangleq \begin{cases} m^2 + \lambda, & \text{if } k = n, \\ \mathsf{E}\left[\min\limits_{u\in\{\pm 1\}} q_{k+1}\left(m + u, \lambda + 2uRU_1\sqrt{\lambda} + R^2\right)\right], & \text{if } 1 \le k < n. \end{cases} \tag{5.10}$$

[8] Note that under our assumption, it is easy to verify that each covariate vector $Z_{k,2:p}$ is zero mean. Our choice of normalization $\mathsf{E}[R^2] = p - 1$ ensures that the covariance matrix of $Z_{k,2:p}$ is given by Σ. This second moment requirement does exclude heavy-tailed elliptical distributions such as the Cauchy distribution. However, it is necessary so that our performance criteria (expected precision) is finite.

Here, when $k < n$, the expectation is taken over independent random variables U and R that are the random variables in the stochastic decomposition of $Z_{1,2:p}$ from Assumption 5.4.6. Then,

1. At each time k, the optimal continuation cost for the dynamic program (P3′) is given by

$$Q_k(\delta_k, \Delta_k) = q_k\left(\delta_k, \|\Delta_k\|^2_{\Sigma^{-1}}\right).$$

   In other words, this is the expected terminal cost, given then covariates observed and the allocations made up to and including time k, assuming optimal decisions are made at all future times.

2. Suppose the allocation x^*_k at each time k is made according to

$$x^*_k \in \operatorname*{arg\,min}_{u\in\{\pm 1\}} \; q_k\left(\delta_{k-1} + u, \|\Delta_{k-1} + uZ_{k,2:p}\|^2_{\Sigma^{-1}}\right). \tag{5.11}$$

   Then, the sequence of allocations x^* is optimal for the online experiment design problem (P3′).

For the case of Gaussian distribution, the recursion (5.10) for solving the DP can be simplified according to the following corollary:

Corollary 2. *If Assumption 5.3.3 holds, then, for $1 \le k \le n$, the functions $q^{\text{gauss}}_k : \mathbb{Z} \times \mathbb{R}_+ \to \mathbb{R}$ are given by*

$$q^{\text{gauss}}_k(m, \lambda) \triangleq \begin{cases} m^2 + \lambda, & \text{if } k = n, \\ \mathsf{E}\left[\min\limits_{u\in\{\pm 1\}} \; q^{\text{gauss}}_{k+1}\left(m + u, (\sqrt{\lambda} + u\eta)^2 + \xi\right)\right], & \text{if } 1 \le k < n. \end{cases} \tag{5.12}$$

Here, when $k < n$, the expectation is taken over independent random variables $(\eta, \xi) \in \mathbb{R}^2$, where $\eta \sim N(0,1)$ is a standard normal random variable, and $\xi \sim \chi^2_{p-2}$ is chi-squared random variable with $p-2$ degrees of freedom.[9]

We provide the proofs for Theorem 5.4.7 and Corollary 2 in the Electronic Companion to this paper. We make the following observations:

1. A key point is that, unlike the standard dynamic programming decomposition of Proposition 5.4.3, Theorem 5.4.7 provides a *tractable* way to solve the surrogate problem (P3′), independent of the covariate dimension p. This is because the recursion (5.10) yields

[9] If $p = 2$, we take $\xi \triangleq 0$.

a two-dimensional dynamic program. One of the state variables of this program, m, is discrete, taking values on the integers from $-n$ to n. Further, one can show that, with high probability, the second state variable λ is $O(n^2)$ thereby allowing us to discretize the state-space on a two-dimensional mesh. The functions $\{q_k\}$ can be numerically evaluated on this grid via backward induction. Note that since the expectation in (5.10) is over a two-dimensional random variable, it can be computed via numerical integration. Further details of this procedure are given in Section 5.6.

2. Moreover, the functions $\{q_k\}$ do not directly depend on the matrix Σ at all and only indirectly depend on time horizon n through the remaining time $k - n$. In fact, they only depend on the covariate dimension p. For example, in the Gaussian case, this means that if these functions are computed offline, they can subsequently be applied to *all* p-dimensional problem with a Gaussian data distribution.

3. Finally, the algorithm assumes that the covariance matrix Σ is known. This is needed to compute the $\|\cdot\|_{\Sigma^{-1}}$-norm of Δ_k. In practice, Σ may not be known, and may need to be estimated from data. However, observe that Σ depends only on the distribution of covariates across the subject population, not on the outcome of experiments. In the applications we have in mind, there is typically a wealth of information about this population known in advance of the experimental trials. Hence, Σ can be estimated offline even if the number of covariates p is large and the number of experimental subjects n is small.

   For example, in an online advertising setting, and advertiser may want to compare two creatives using A-B testing with a limited number of experimental subjects. In advance of any experiments, the advertiser can use historical data from other trials or market surveys over the same population of subjects to estimate Σ.

5.5. Variations of the Sequential Problem: A Dynamic Programming Framework

The vanilla formulation of the sequential problem (P2) described in Section 5.2.2 solely optimizes statistical efficiency. In reality, a complete framework must allow the designer to model a number of additional constraints relevant to practical implementation, including budgets on allocations to the treatment arm; controlling selection bias in addition to maximizing efficiency; optimally

stopping an experiment if efficiency objectives are met; and so forth. We will establish that the solution approach described in Section 5.4 applies to a substantially more general class of problem than the vanilla problem (P2).

To setup this dynamic programming framework, we introduce a few new concepts:

- We will think of the allocation at time $1 \leq k \leq n$ as a *bias* $v_k \in [0,1]$. Our optimization algorithm will yield the optimal bias at any given point in time, and then we pick an allocation by flipping a coin with this bias, i.e., setting

$$x_k = \begin{cases} +1 & \text{with probability } v_k, \\ -1 & \text{with probability } 1 - v_k. \end{cases} \tag{5.13}$$

  This is the same decision space as in a biased coin design.

- We are given convex stage wise costs, $c\colon [0,1] \to \mathbb{R}$, that are a function of bias. This can capture for instance, the 'cost' of a sample unit; the extent of 'non-randomness' in a given choice of bias, etc.

- The set of permitted bias v_k at any stage $1 \leq k \leq n$ can be constrained to an arbitrary convex set that is itself a function of the state at that time, $\mathcal{V}_k\left(\delta_{k-1}, \|\Delta_{k-1}\|^2_{\Sigma^{-1}}\right) \subset [0,1]$.

- Instead of a fixed time horizon n, we allow the experiment to be stopped early according to a stopping time $1 \leq \tau \leq n$. As we discuss below this allows us to model optimal early stopping based, for instance, on estimating the treatment effect with a desired precision.

Given these concepts, and an arbitrary parameter $\gamma \geq 0$, consider the following generalization of the problem (P3′):

$$\begin{aligned} (\text{P3}'') \triangleq \text{minimize} \quad & \mathsf{E}\left[\delta_\tau^2 + \|\Delta_\tau\|^2_{\Sigma^{-1}} + \gamma \sum_{k=1}^{\tau} c(v_k)\right] \\ \text{subject to} \quad & v_k \in \mathcal{V}_k\left(\delta_{k-1}, \|\Delta_{k-1}\|^2_{\Sigma^{-1}}\right), \qquad \forall\ 1 \leq k \leq n, \\ & v_k \text{ is } \mathcal{F}_k\text{-measurable}, \qquad \forall\ 1 \leq k \leq n. \end{aligned}$$

Following the same arguments as in Section 5.4.4, (P3″) can be solved according to optimal

continuation costs given by the two-dimensional Bellman recursion[10]

$$q_k(m,\lambda) \triangleq \begin{cases} m^2+\lambda, & \text{if } k=\tau, \\ \mathsf{E}\Big[\min\limits_{v\in\mathcal{V}_{k+1}(m,\lambda)} \gamma c(v) & \\ \quad +vq_{k+1}\left(m+1,\lambda+2RU_1\sqrt{\lambda}+R^2\right) & \text{if } 1\le k<\tau, \\ \quad +(1-v)q_{k+1}\left(m-1,\lambda-2RU_1\sqrt{\lambda}+R^2\right)\Big], & \end{cases} \tag{5.14}$$

for each time k. Given the optimal continuation costs, an optimal decision v_k at each time k can be computed according to

$$\begin{aligned} v_k^* \in \underset{v\in\mathcal{V}_k\left(\delta_{k-1},\|\Delta_{k-1}\|^2_{\Sigma^{-1}}\right)}{\arg\min} & \gamma c(v) \\ & + vq_k\left(\delta_{k-1}+1,\|\Delta_{k-1}+Z_{k,2:p}\|^2_{\Sigma^{-1}}\right) \\ & + (1-v)q_k\left(\delta_{k-1}-1,\|\Delta_{k-1}-Z_{k,2:p}\|^2_{\Sigma^{-1}}\right), \end{aligned} \tag{5.15}$$

In the following, we illustrate how (P3″) addresses several practical variations of the sequential allocation problem:

Selection Bias. An important consideration that has emerged in the literature on A-B testing is managing so-called 'selection bias'. Following Blackwell and Hodges (1957), one commonly defines the selection bias of an allocation over n time steps as $\frac{2}{n}\sum_{k=1}^n |v_k - 1/2|$. Notice that perfect randomization has zero selection bias, whereas a fully deterministic procedure (where v_k is either 0 or 1) has the highest bias possible, one.

It is frequently important to balance this bias against efficiency (or, equivalently, loss). In particular, we want a Pareto optimal solution across the two criteria. Atkinson (2014) compares a multitude of state-of-the-art biased coin design (BCD) procedures and calls a procedure 'admissible' if it is not Pareto dominated by some other procedure. He finds that none of the heuristics he examines can be ruled out implying that *none* of these heuristics are Pareto optimal. But by varying $\gamma \ge 0$ in (P3″), we can generate a Pareto optimal solution at any point on the trade-off curve. Specifically, to incorporate selection bias into our framework, we simply define

$$c(v) \triangleq |v-1/2|, \quad \tau \triangleq n, \quad \mathcal{V}_k \triangleq [0,1]. \tag{5.16}$$

[10] In order for the decomposition (5.14) to apply, an additional technical assumption is needed on the stopping time τ: we assume that, for each $1 \le k < \tau$, the distribution of the random variable corresponding the to future stopped payoff $\delta_\tau^2 + \|\Delta_\tau\|^2_{\Sigma^{-1}}$ is conditionally independent of the history given the current state (δ_k, Δ_k).

Our approach can consequently produce any design on the Pareto frontier, and thus Pareto dominate state-of-the-art BCD designs. We will see this numerically in Section 5.6.

Notice that the optimal policy equation (5.15) in the setting of (5.16) is a linear program. Direct examination of this program yields an interesting insight: at every time k, the optimal action for (P3″) is restricted to $v_k \in \{0, 1/2, 1\}$. In other words, an optimal policy will only either take a deterministic action or fully randomize. This is in contrast to the main BCD heuristics developed in the literature (some of which we will describe shortly in Section 5.6.3), which tend to vary probabilities over the entire interval $[0, 1]$.

Allocation Budget. Assuming a test with a total sample size of n, the designer may be happy to assign these samples to the control arm (the 'status quo') but may want to limit exposure to the test. Formally, we may want to have a budget B on the number of $+1$ allocations in the trial. As it turns out BCD does not naturally extend to this setting (Han et al. 2009, Kuznetsova and Tymofyeyev 2012). (P3″) can trivially incorporate a budget constraint, we simply define

$$c(v) \triangleq 0, \quad \tau \triangleq n, \quad \mathcal{V}_k\left(\delta_{k-1}, \|\Delta_{k-1}\|^2_{\Sigma^{-1}}\right) \triangleq \begin{cases} [0,1] & \text{if } k + \delta_{k-1} < 2B, \\ \{0\} & \text{otherwise.} \end{cases}$$

Endogenous Stopping. Consider the (not uncommon) scenario where there is an economic cost associated with every incremental sampling unit in a sequential trial, and all we care about is estimating the treatment effect up to a desired level of precision; see Johari et al. (2017) for a broader discussion of related problems. In such a scenario, we may opportunistically want to stop early so that the sample size is in fact picked endogenously. For concreteness, let us suppose that the unit cost per sample is a constant r. Assume further that it suffices to estimate the treatment effect with precision κ, unless the trial has run up to a sample size of n in which case we must stop. One can think of n here as an upper bound on sample size imposed by the trial designer. The objective is simply to minimize the expected cost of the trial. This problem is easily modeled in our framework. Specifically, (P3″) can capture this problem by defining

$$c(v) \triangleq r, \quad \tau \triangleq \min\left\{k \geq 1 \ : \ k - \frac{1}{k}\left(\delta_k^2 + \|\Delta_k\|^2_{\Sigma^{-1}}\right) \geq \kappa\sigma^2\right\} \wedge n, \quad \mathcal{V}_k \triangleq [0,1].$$

5.6. Experiments

This section focuses on numerical experiments with data. We will attempt to highlight the relative merits of our approach vis-à-vis simple randomization, as well as biased coin designs (BCDs). As discussed in the literature review, BCDs are an approach to minimizing loss (or equivalently, maximizing efficiency) by dynamically adjusting for covariate imbalances.

Our goal will be to show that for a given level of *selection bias*, our approach provides an improvement in efficiency (or a reduction in loss) over competing BCDs. Equivalently, our approach can achieve a given level of efficiency with a smaller level of selection bias. We will study these relative merits for varying values of sample size n, and the number of covariates p. Finally, while our analysis in Section 5.4 required the covariates to follow an elliptical distribution, such a requirement may not hold in real applications. As such we conduct experiments using click log data from Yahoo! wherein the covariates are categorical; we show that our approach enjoys similar relative merits in this setting.

5.6.1. BCDs, Loss, and Selection Bias

Let $v_k \in [0,1]$ denote the probability that the kth allocation is set to $x_k = +1$ under a given allocation rule $\mathcal{A}$. Recall from Section 5.5 that a measure of selection bias under $\mathcal{A}$ is defined according to

$$\text{Bias}_{\mathcal{A}} \triangleq \mathsf{E}\left[\frac{2}{n}\sum_{k=1}^{n}|v_k - 1/2|\right] \in [0,1].$$

(Here, we have normalized the bias to be contained in the unit interval.) This measure captures the extent of randomness (or, equivalently, how predictable any given allocation is) under $\mathcal{A}$ (Blackwell and Hodges 1957). Also, recall our definition of loss,

$$\text{Loss}_{\mathcal{A}} \triangleq n - \mathsf{E}\left[x^\top P_{Z^\perp} x\right] = \mathsf{E}\left[x^\top Z\left(Z^\top Z\right)^{-1} Z^\top x\right] \geq 0.$$

The loss under $\mathcal{A}$ is interpreted as the effective number of samples on which information is lost due to an imbalance in covariates. It is well known that any allocation rule engenders a trade-off between loss and selection bias, so that a comparison between rules ideally compares the entire trade-off curve attained by the two rules (Atkinson 2002). We will do precisely this in the experiments that follow.

Observe that the expressions for bias and loss do not depend on the experimental outcomes

$\{y_k\}$. From an empirical perspective, this is helpful: we can assess any rule $\mathcal{A}$, given only access to the covariate distribution. The conclusions we draw on the relative merits of one approach with respect to another hold across any linear model for the given covariate structure.

5.6.2. Data

We run our experiments on two different data distributions for the covariates. Assumption 5.4.1 holds in both cases. Thus, $\{Z_k\}$ are i.i.d. and $Z_{k,1}$ is assumed to be 1. We run our experiments with the following sampling distributions for $Z_{2:p}$:

Synthetic Gaussian Data. In our synthetic experiments, we assume that $Z_{2:p}$ follows multivariate normal distribution. This is, of course, an elliptical distribution, so that Assumption 5.3.3 is satisfied. For the covariance matrix Σ, we set $\Sigma_{ii} = 1.0$ and $\Sigma_{ij} = 0.1$ for any $j \neq i$.

Yahoo! User Data. To experiment on data from a more realistic setting, we use a dataset of user click log data from the Yahoo! front page.[11] The users here are visitors to 'Featured Tab of the Today Module' on the Yahoo! front page. In the dataset, each user has 136 associated features, such as age and gender. Each feature is binary, taking values in $\{0, 1\}$. Some of these features were constant throughout the dataset, and these were discarded. Duplicate and co-linear features were discarded as well. Features were selected at random until up to $p = 40$ features were collected. Feature selection was repeated independently in each simulation trial.

Our algorithm requires the covariance matrix of the data as an input. For this purpose, we estimate the covariance matrix from a portion of the dataset. This estimate is obtained by simply taking a sample average across 1 million data points kept aside from the rest of the experiments.

Finally, for evaluation purposes, we require a generative model for the data. To this end, from a set of 1 million data points we sample individual data points, with replacement. In other words, as the sampling distribution we use the empirical distribution of the 1 million data points used for testing. Such a sampling procedure is intended to mimic the arrival of users on the Yahoo! front page.

[11] This dataset is obtained from the Yahoo! Labs repository of datasets available for academic research, and can be downloaded as "R6B — Yahoo! Front Page Today Module User Click Log Dataset, version 2.0" at http://webscope.sandbox.yahoo.com/catalog.php?datatype=r.

5.6.3. Algorithms

Dynamic Programming (Our Approach). The problem at hand is addressed by the dynamic programming formulation described in Sections 5.5. As such, we are required to compute the 2-dimensional value functions given by $\{q_k\}_{1\leq k\leq n}$. These functions are computed offline by backward induction following (5.14). Here, we provide the computational details for this operation. In particular, given $q_{k+1}(\cdot,\cdot)$, we compute $q_k(\cdot,\cdot)$ as follows:

1. Discretization: The first state variable m is discrete and can take values from $-n$ to n. We discretize values for the second state variable λ on a geometric mesh taking values λ_0^i for $\lambda_0 \triangleq 1.5$ and $0 \leq i \leq 26$. The maximum value value of λ was chosen so that $\|\Delta_k\|^2_{\Sigma^{-1}}$ has a low probability of exceeding it.

2. Sampling: For each discretized pair (m,λ) we estimate $q_k(m,\lambda)$ via Monte Carlo simulation. In particular, $N = 10{,}000$ pairs[12] $(\xi,\eta) \in \mathbb{R}^2$ are sampled from the appropriate distributions and $q_k(m,\lambda)$ is estimated according to (5.14) using the corresponding empirical measure. We use the same sample set of (ξ,η) for all (m,λ) at which this is evaluated.

3. Interpolation: Given an (m,λ) such that λ is not a discretized mesh point, we estimate $q_{k+1}(m,\lambda)$ in the Bellman recursion (5.14) by linear interpolation between the closest points in the discretized mesh.

Biased Coin Designs. In addition to our own dynamic programming algorithm, we will consider several other rules proposed in the literature. These include: Rule ABCD (Baldi Antognini and Zagoraiou 2011), which following Atkinson (2014), we refer to as Rule J; Smith's rule (Rule S) (Smith 1984b,a); Atkinson's rule (Rule A) (Atkinson 1982), and the Bayesian procedure of Ball et al. (1993) (Rule B). Rules J, S, and B are all parameterized by a scalar parameter, which we denote ρ, that may take values in $(0,\infty)$. Rule A is a special case of Rule S taking $\rho = 1$. As $\rho \to 0$, these rules become equivalent to randomization. On the other hand, as $\rho \to \infty$, these rules become entirely deterministic in nature. As such, for values of ρ close to zero, one expects low selection bias whereas as $\rho \to \infty$ one expects to see a reduction in loss at the expense of selection bias; a deterministic rule has the largest possible selection bias of 1. In order to

[12] In all examples, our algorithm assumes that the covariate data is generated from a multivariate normal, even when this was not true (Yahoo! dataset). In this case, when $Z_{2:p} \sim N(I,\Sigma)$ is multivariate normal, $\lambda + 2uRU_1\sqrt{\lambda} + R^2$ has the same distribution as $(\sqrt{\lambda}+u\eta)^2+\xi$ where η is a standard normal and ξ is a chi-squared random variable with $p-2$ degrees of freedom. See also Corollary 2.

precisely specify each of these rules, define

$$d_k\left(u_{k+1}, Z_{k+1,2:p}\right) \triangleq \left(1 - u_{k+1}\delta_k/k - u_{k+1}Z_{k+1,2:p}^\top \Sigma^{-1}\Delta_k/k\right)^2$$

where $u_{k+1} \in \{\pm 1\}$, $Z_{k+1,2:p} \in \mathbb{R}^{p-1}$, and δ_k and Δ_k have the usual definitions (5.4). For background on the function $d_k(\cdot,\cdot)$, see Atkinson (1982); this quantity arises naturally in the sequential design of D_A-optimal experiments. The rules described above then take the following form:

1. **Rules S/A**: Assign $x_{k+1} = +1$ with probability

$$v_{k+1} \triangleq \frac{d_k(+1, Z_{k+1,2:p})^\rho}{d_k(+1, Z_{k+1,2:p})^\rho + d_k(-1, Z_{k+1,2:p})^\rho}.$$

   The parameter ρ can take values in $(0, \infty)$. Rule A corresponds to the special case where $\rho = 1$.

2. **Rule B**: Assign $x_{k+1} = +1$ with probability

$$v_{k+1} \triangleq \frac{(1 + d_k(+1, Z_{k+1,2:p}))^\rho}{(1 + d_k(+1, Z_{k+1,2:p}))^\rho + (1 + d_k(-1, Z_{k+1,2:p}))^\rho}.$$

   The parameter ρ can again take values in $(0, \infty)$. This rule is very similar to Rule S, but permits a Bayesian interpretation (Ball et al. 1993).

3. **Rule D**: Assign $x_{k+1} = +1$ deterministically if $d_k(+1, Z_{k+1,2:p}) > d_k(-1, Z_{k+1,2:p})$, set $x_{k+1} = -1$ otherwise. This rule is obtained in the limit as $\rho \to \infty$ for rules A, S, and B. Note that this deterministic rule is equivalent to a *myopic* policy that seeks to optimize the objective of (P3′) assuming that x_{k+1} is the final allocation to be made, and ignoring the impact of this allocation on future decision making.

4. **Rule J**: Define the 'discrepancy' after k allocations, $D_k(Z_{k+1,2:p})$ according to

$$D_k(Z_{k+1,2:p}) \triangleq \frac{2 - k\left(d_k(+1, Z_{k+1,2:p}) + d_k(-1, Z_{k+1,2:p})\right)}{d_k(+1, Z_{k+1,2:p}) - d_k(-1, Z_{k+1,2:p})},$$

   assuming $d_k(+1, Z_{k+1,2:p}) \neq d_k(-1, Z_{k+1,2:p})$. If $D_k(Z_{k+1,2:p}) < 0$, we assign $x_{k+1} = +1$ with probability

$$v_{k+1} \triangleq \frac{|D_k(Z_{k+1,2:p})|^\rho}{1 + |D_k(Z_{k+1,2:p})|^\rho}.$$

If, on the other hand $D_k(Z_{k+1,2:p}) > 0$, we assign $x_{k+1} = +1$ with probability

$$v_{k+1} \triangleq \frac{1}{1 + |D_k(Z_{k+1,2:p})|^\rho}.$$

Finally, if $D_k(Z_{k+1,2:p}) = 0$ or $d_k(+1, Z_k) = d_k(-1, Z_k)$, we simply randomize ($v_{k+1} = 1/2$). The parameter ρ can again take values in $(0, \infty)$.

5.6.4. Results

Our goal is to compare the statistical efficiency of our dynamic programming-based sequential algorithm to the various competing BCDs discussed above while controlling for selection bias. In order to do this, we run each BCD procedure for an increasing sequence of value of ρ. The smallest value used, $\rho = 0$, is simply equivalent to randomized allocation. The largest value of ρ we considered for each scheme was chosen so that the rule was effectively deterministic. We implemented our sequential DPalgorithm for an increasing sequence of values of γ, tracing out a similar trade-off curve.

Results are reported in Figures 5.1, 5.2, and 5.3. Of these, Figures 5.1 and 5.2 show results on synthetic Gaussian data while Figure 5.3 shows results on the Yahoo! dataset. Each data point in these figures is the average of 10,000 independent Monte Carlo trials with shared randomness across all BCD rules and our own rule; and different data points were generated for each rule by varying their respective configurations of ρ and γ.

These figures reveal that:

1. For any target level of selection bias, our dynamic programming algorithm has the smallest loss among all of the alternatives implemented. In this way, the DP approach Pareto dominates all alternatives. The relative improvement in loss can be non-trivial: the loss incurred under our approach can be up to five times smaller for moderate budgets on selection bias. Put a different way the effective number of samples 'lost' due to covariate imbalance can be substantially smaller for a given budget on selection bias.

2. The relative improvement alluded to above is particularly pronounced for smaller values of p/n. Our intuition here is as follows: keeping n fixed one expects to require fewer non-random allocations for small p. As such, the importance of strategizing on *when* to employ a non-random allocation has greater impact in such a setting.

3. The relative merits of our sequential approach appear more pronounced in the setting where n is larger.

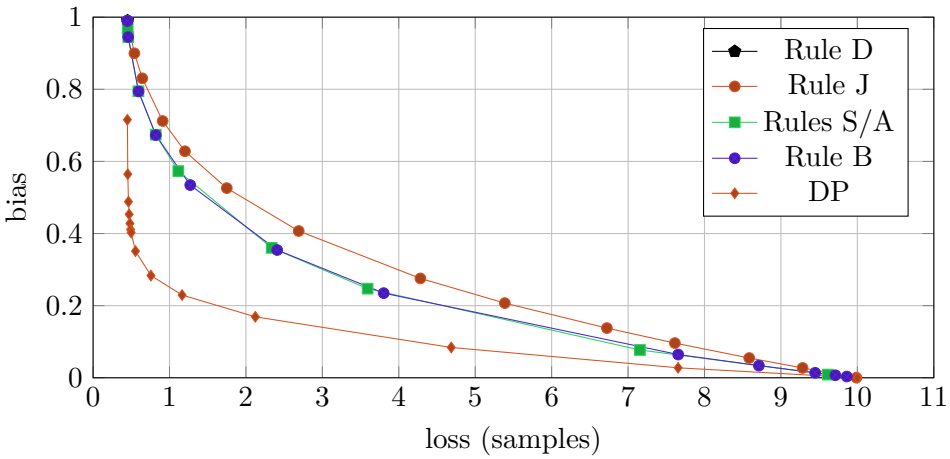
4. Finally, observe that Figure 5.3 shows results on the Yahoo! dataset, and that the covariates in this experiment are in fact categorical. Despite this we see that our approach exhibits similar improvements relative to the competing BCD schemes.

5.7. Conclusion

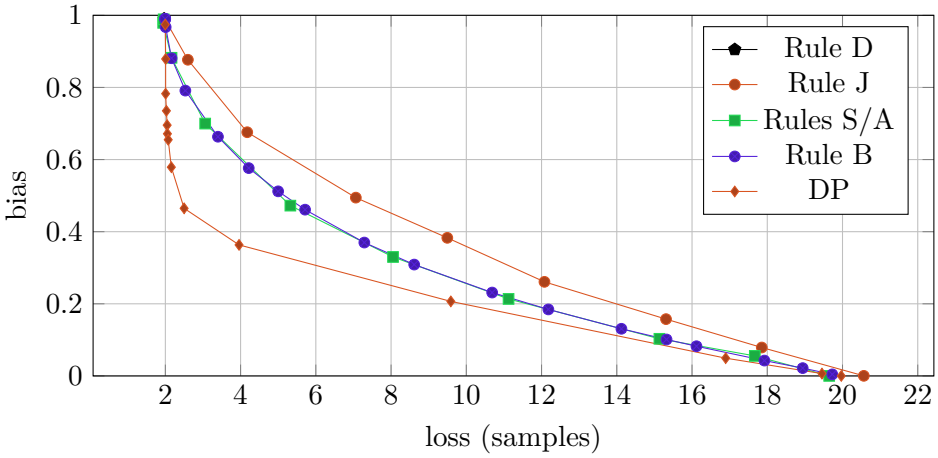
We conclude with a summary of what we have accomplished and what we view as key directions for further research. At a conceptual level, this paper illustrates the power of the 'optimization' viewpoint in what are inherently statistical problems: we have presented a provably near optimal solution to a problem for which a plethora of heuristics were available. In addition to establishing the appropriate approach to this problem, the algorithms we have developed are eminently practical and easy to implement — a property that is crucial for the sorts of applications that motivated this work. On a more pragmatic note, we have quantified the *value* of these sorts of optimization approaches establishing precise estimates of the benefits optimization approaches provide over straightforward randomization. These estimates illustrate that in so-called high dimensional setting — i.e., in settings where the number of covariates is large, such approaches can provide order of magnitude improvements in sampling efficiency.

Our progress does come at the expense of structural assumptions on the relationship between the observed effect and observable covariates. In particular, we assumed a linear model with exogenous noise. Any such structural assumption is restrictive. In the event that these assumptions fail, they could result in biased estimates of the treatment effect. With that said, it appears difficult to overcome the risk of such a bias while using a covariate dependent treatment assignment scheme. In addition to these structural assumptions, we also required that the experiment designer have some knowledge on the distribution of the covariates (their covariance matrix). Our theoretical results made further distributional assumptions on these covariates. Much remains to be done to mitigate the impact of these limiting assumptions, and as such a number of directions remain for future research. We highlight several here in parting:

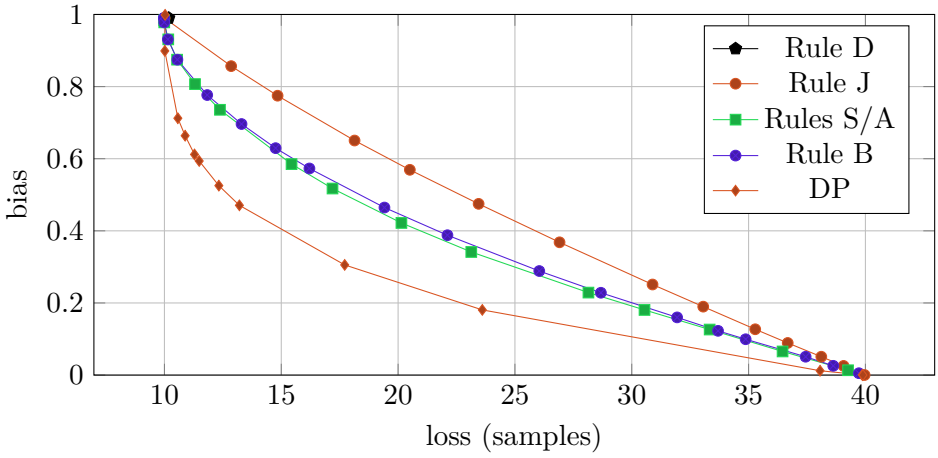
1. Normality: To what extent can our assumption on the normality of covariates be relaxed? Can we develop approximation guarantees for the situation when covariates are not normally distributed?



(a) $p = 10$

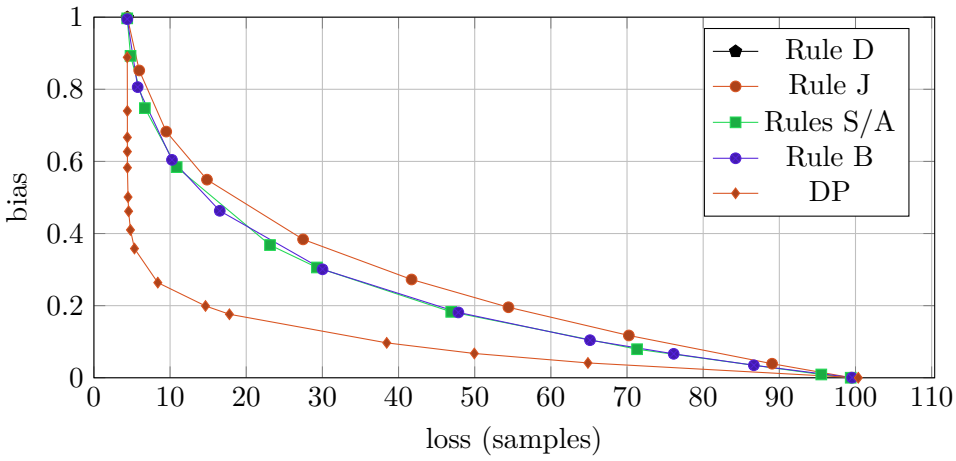


(b) $p = 20$

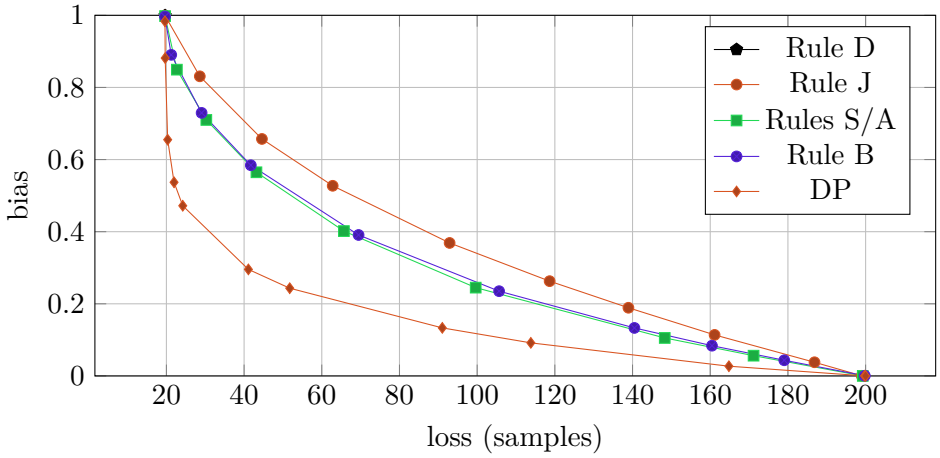


(c) $p = 40$

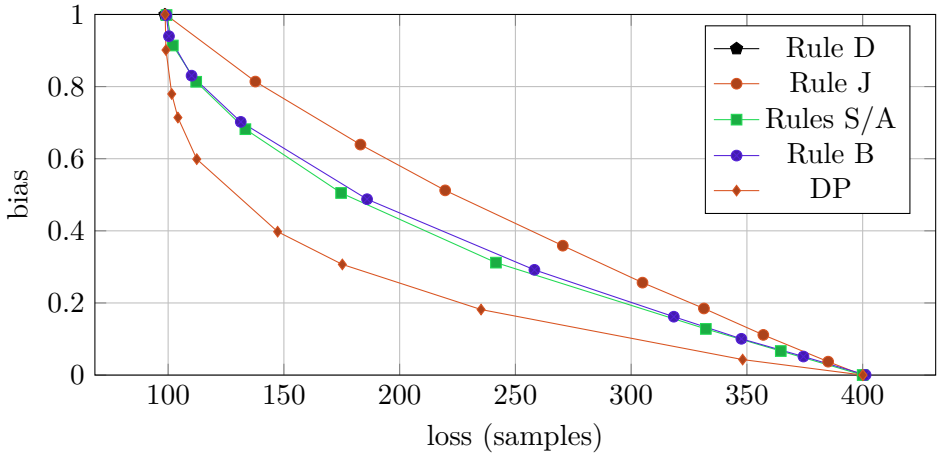
Figure 5.1: Bias-loss trade-off on synthetic Gaussian data for $n = 100$ and varying values of p.



(a) $p = 100$

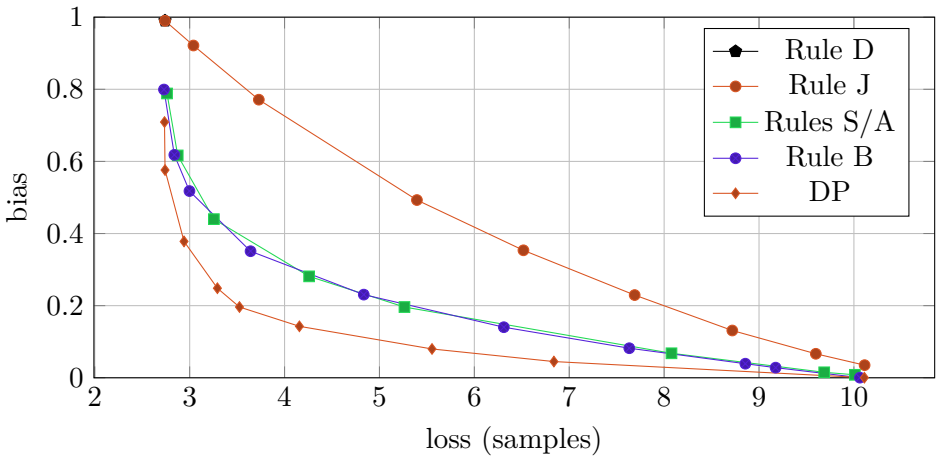


(b) $p = 200$

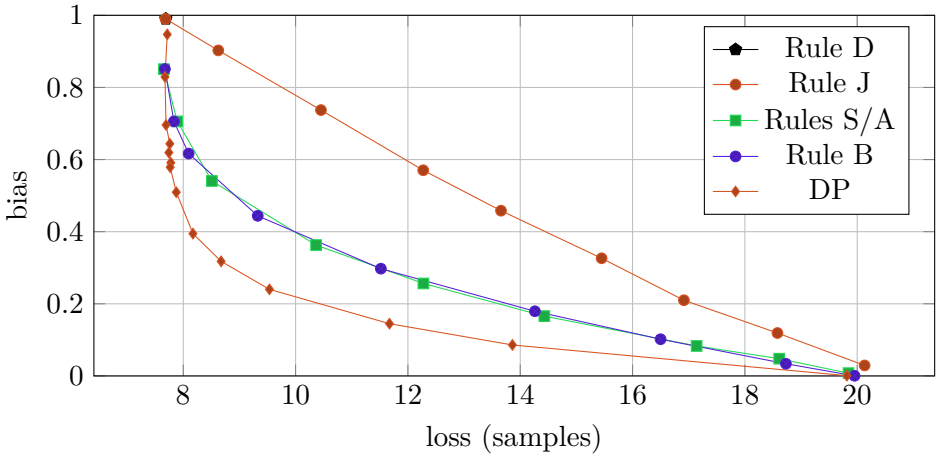


(c) $p = 400$

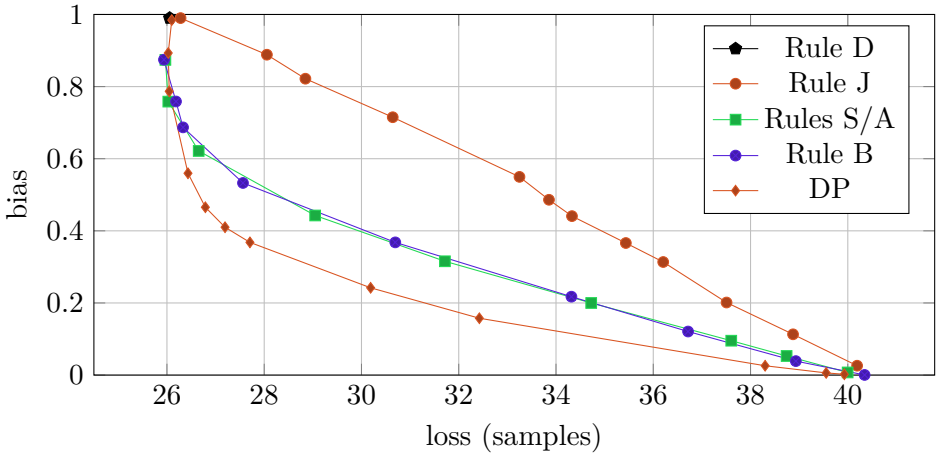
Figure 5.2: Bias-loss trade-off on synthetic Gaussian data for $n = 1000$ and varying values of p.



(a) $p = 10$



(b) $p = 20$



(c) $p = 40$

Figure 5.3: Bias-loss trade-off on the Yahoo! dataset for $n = 100$ and varying values of p.

2. Non-linear models: Can we allow for a nonlinear dependence on covariates? One direction to accomplish this is perhaps a reliance of some manner of non-parametric 'kernel' approach. The good news here is that the value of optimization is likely to be even higher in such an infinite-dimensional setting.

3. More than two alternatives: The present paper considers only the two alternative setting, an important direction for future work would be to consider settings where there is a larger number of choices.

Chapter 6

Conclusion

In this thesis, we considered various challenges that are faced by online platforms today. We showed how these platforms can benefit from systematic algorithmic approaches to these problems. In particular, our contributions can be summarized as:

- We proposed a multi-armed bandit framework for modeling the SMS routing problem and presented a near-optimal algorithm for the problem.
- We proposed a sub-linear time algorithm for making personalized recommendations to users in real-time.
- We proposed a choice model and an assortment optimization algorithm to incorporate the purchase of multiple products in a single transaction.
- We proposed an efficient algorithm for allocating sequentially arriving users to test and control groups for performing A-B testing on online platforms.

Appendix A

Appendix to Chapter 2

Outline The appendix of this chapter is organized as follows.

- Section A.1 contains technical lemmas used in subsequent proofs.
- Section A.2 contains a proof of the lower bound.
- Section A.3 contains proofs related to the performance of various algorithms presented in the paper.
- Section A.4 gives a detailed description of the CS-ETCalgorithm when the costs of the arms are unknown and random.

A.1. Technical Lemmas

Lemma A.1.1 (Taylor's Series Approximation)**.** For $x > 0$, $\ln(1+x) \geq x - \frac{x^2}{1-x^2}$.

Proof. For $x > 0$,

$$\begin{aligned}
\ln(1+x) &= x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots \\
&\geq x - \frac{x^2}{2} - \frac{x^4}{4} - \cdots \qquad \text{(because } x > 0\text{)} \\
&\geq x - x^2 - x^4 \\
&= x - x^2(1 + x^2 + x^4 + \cdots) \\
&= x - \frac{x^2}{1-x^2}.
\end{aligned}$$

■

Lemma A.1.2 (Taylor's Series Approximation)**.** For $x > 0$, $\ln(1-x) \geq -x - \frac{x^2}{1-x}$.

Proof. For $x > 0$,

$$\begin{aligned}
\ln(1-x) &= -x - \frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} + \cdots \\
&\geq -x - x^2 - x^3 - x^4 - \cdots \qquad \text{(because } x > 0\text{)} \\
&= -x - x^2(1 + x + x^2 + \cdots) \\
&= -x - \frac{x^2}{1-x}.
\end{aligned}$$

■

Lemma A.1.3 (Pinsker's inequality)**.** Let $Ber(x)$ denote a Bernoulli distribution with mean x where $0 \leq x \leq 1$. Then, $KL(Ber(p); Ber(p+\epsilon)) \leq \frac{4\epsilon^2}{p}$ where $0 < p \leq \frac{1}{2}$, $0 < \epsilon \leq \frac{p}{2}$ and $p + \epsilon < 1$ and the KL divergence between two Bernoulli distributions with mean x and y is given as $KL(Ber(x); Ber(y)) = x \ln \frac{x}{y} + (1-x) \ln \frac{1-x}{1-y}$.

Proof. $KL(Ber(p); Ber(p+\epsilon)) = p \ln \frac{p}{p+\epsilon} + (1-p) \ln \frac{1-p}{1-p-\epsilon}$

$$\begin{aligned}
KL(Ber(p); Ber(p+\epsilon)) &= p \ln \frac{p}{p+\epsilon} + (1-p) \ln \frac{1-p}{1-p-\epsilon} \\
&= -p \ln \left(1 + \frac{\epsilon}{p}\right) - (1-p) \ln \left(1 - \frac{\epsilon}{1-p}\right) \\
&\leq -p \left(\frac{\epsilon}{p} - \frac{\frac{\epsilon^2}{p^2}}{1 - \frac{\epsilon^2}{p^2}}\right) - (1-p) \left(-\frac{\epsilon}{1-p} - \frac{\frac{\epsilon^2}{(1-p)^2}}{1 - \frac{\epsilon}{1-p}}\right)
\end{aligned}$$

(Using Lemmas A.1.1 and A.1.2.)

Thus,

$$\begin{aligned}
KL(Ber(p); Ber(p+\epsilon)) &\leq -\epsilon + \frac{\epsilon^2}{p\left(1-\frac{\epsilon^2}{p^2}\right)} + \epsilon + \frac{\epsilon^2}{(1-p)\left(1-\frac{\epsilon}{1-p}\right)} \\
&\leq \frac{\epsilon^2}{p\left(1-\frac{1}{4}\right)} + \frac{\epsilon^2}{(1-p)\left(1-\frac{1}{2}\right)} \qquad \left(\text{because } \frac{\epsilon}{1-p} \leq \frac{\epsilon}{p} \leq \frac{1}{2}\right) \\
&= \frac{4\epsilon^2}{3p} + \frac{2\epsilon^2}{1-p} \\
&\leq \frac{2\epsilon^2}{p} + \frac{2\epsilon^2}{1-p} \\
&\leq \frac{2\epsilon^2}{p} + \frac{2\epsilon^2}{p} \qquad (\text{because } p \leq \frac{1}{2}) \\
&= \frac{4\epsilon^2}{p}.
\end{aligned}$$

■

A.2. Proof of Lower Bound

Proof of Lemma 2.3.3. In the family of instances $\Phi_{\theta,p,\epsilon}$, the costs of the arms are same across instances. Arm 0 is the cheapest arm in all the instances. With this, we define a modified notion of quality regret which penalizes the regret only when this cheap arm is pulled as

$$\mathsf{Mod\_Quality\_Reg}_{\pi}(T, \alpha, \boldsymbol{\mu}, \boldsymbol{c}) = \sum_{t=1}^{T} \max\{\mu_{m*} - \mu_{\pi_t}, 0\} \mathbb{1}(c_{i_t} = 0). \tag{A.1}$$

An equivalent notation for denoting the modified regret of policy π on an instance I of the problem is $\mathsf{Mod\_Quality\_Reg}_{\pi}(T, \alpha, I)$. This modified quality regret is at most equal to the quality regret. For proving the lemma, we will show a stronger result that there exists an instance $\phi_{0,p,\epsilon}$ such that $\mathsf{Mod\_Quality\_Reg}(T, 0, \phi_{0,p,\epsilon}) + \mathsf{Cost\_Reg}(T, 0, \phi_{0,p,\epsilon})$ is $\Omega\left(pK^{\frac{1}{3}}T^{\frac{2}{3}}\right)$ which will imply the required result.

Let us first consider any deterministic policy (or algorithm) π. For a deterministic algorithm, the number of times an arm is pulled is a function of the observed rewards. Let the number of times arm j is played be denoted by N_j and let the total number of times any arm with cost 1 i.e. an expensive arm is played be $N_{exp} = 1 - N_0$. For any a such that $1 \leq a \leq K$, we can use the proof of Lemma A.1 in Auer et al. (2002a), with function $f(\mathbf{r}) = N_{exp}$ to get

$$\mathbb{E}^a[N_{exp}] \leq \mathbb{E}^0[N_{exp}] + 0.5T\sqrt{2\mathbb{E}^0[N_a]KL(Ber(p); Ber(p+\epsilon))}$$

where $\mathbb{E}^j$ is the expectation operator with respect to the probability distribution defined by the random rewards in instance $\Phi^j_{0,p,\epsilon}$. Thus, using Lemma A.1.3, we get,

$$\mathbb{E}^a\left[N_{exp}\right] \leq \mathbb{E}^0\left[N_{exp}\right] + 0.5T\sqrt{\mathbb{E}^0[N_a]8\epsilon^2/p}. \tag{A.2}$$

Now, let us look at the regret of the algorithm for each instance in the family $\Phi_{0,p,\epsilon}$. We have

1. $\mathsf{Cost\_Reg}_\pi(T,\alpha,\Phi^0_{0,p,\epsilon}) = \mathbb{E}^0[N_{exp}],\ \mathsf{Mod\_Quality\_Reg}_\pi(T,\alpha,\Phi^0_{0,p,\epsilon}) = 0$
2. $\mathsf{Cost\_Reg}_\pi(T,\alpha,\Phi^a_{0,p,\epsilon}) = 0,\ \mathsf{Mod\_Quality\_Reg}_\pi(T,\alpha,\Phi^a_{0,p,\epsilon}) = \epsilon\left(T - \mathbb{E}^a[N_{exp}]\right).$

Now, define randomized instance $\phi_{0,p,\epsilon}$ as the instance obtained by randomly choosing from the family of instances $\Phi_{0,p,\epsilon}$ such that $\phi_{0,p,\epsilon} = \Phi^0_{0,p,\epsilon}$ with probability $1/2$ and $\phi_{0,p,\epsilon} = \Phi^a_{0,p,\epsilon}$ with probability $1/2K$ for $1 \leq a \leq K$. The expected regret of this randomized instance is

$$\begin{aligned}
&\mathbb{E}\left[\mathsf{Mod_Quality_Reg}_\pi(T,0,\phi_{0,p,\epsilon}) + \mathsf{Cost_Reg}_\pi(T,0,\phi_{0,p,\epsilon})\right] \\
&= \frac{1}{2}\left(\mathsf{Mod_Quality_Reg}_\pi(T,\alpha,\Phi^0_{0,p,\epsilon}) + \mathsf{Cost_Reg}_\pi(T,\alpha,\Phi^0_{0,p,\epsilon})\right) + \\
&\quad \frac{1}{2K}\sum_{a=1}^{K}\left(\mathsf{Mod_Quality_Reg}_\pi(T,\alpha,\Phi^a_{0,p,\epsilon}) + \mathsf{Cost_Reg}_\pi(T,\alpha,\Phi^a_{0,p,\epsilon})\right) \\
&= \frac{1}{2}\mathbb{E}^0[N_{exp}] + \frac{1}{2K}\sum_{a=1}^{K}\epsilon(T - \mathbb{E}^a[N_{exp}]) \\
&\geq \frac{1}{2}\mathbb{E}^0[N_{exp}] + \frac{1}{2K}\sum_{a=1}^{K}\epsilon\left(T - \mathbb{E}^0\left[N_{exp}\right] - \frac{1}{2}T\sqrt{\mathbb{E}^0[N_a]\frac{8\epsilon^2}{p}}\right) \qquad \text{(using (A.2))} \\
&= \frac{1}{2}\left[\epsilon T + (1-\epsilon)\sum_{a=1}^{K}\mathbb{E}^0[N_a] - \frac{T\epsilon}{2K}\sum_{a=1}^{K}\sqrt{\frac{8\epsilon^2}{p}\mathbb{E}^0[N_a]}\right] \\
&= \frac{1}{2}\sum_{a=1}^{K}\left[\frac{\epsilon T}{K} + (1-\epsilon)(\mathbb{E}^0[N_a])^2 - T\mathbb{E}^0[N_a]\epsilon^2\frac{\sqrt{2}}{K\sqrt{p}}\right] \\
&= \frac{1}{2}\sum_{a=1}^{K}\left[\left(\sqrt{1-\epsilon}\mathbb{E}^0[N_a] - \frac{\epsilon^2 T}{2K}\sqrt{\frac{2}{p(1-\epsilon)}}\right)^2 + \frac{\epsilon T}{K} - \frac{\epsilon^4 T^2}{2pK^2(1-\epsilon)}\right] \\
&\geq \frac{1}{2}\sum_{a=1}^{K}\frac{\epsilon T}{K} - \frac{\epsilon^4 T^2}{2pK^2(1-\epsilon)} \\
&= \frac{\epsilon T}{2} - \frac{\epsilon^4 T^2}{4pK(1-\epsilon)}
\end{aligned}$$

Taking $\epsilon = \frac{p}{2}(\frac{K}{T})^{\frac{1}{3}}$, we get $\mathbb{E}\left[\mathsf{Mod\_Quality\_Reg}_\pi(T,0,\phi_{0,p,\epsilon}) + \mathsf{Cost\_Reg}_\pi(T,0,\phi_{0,p,\epsilon})\right]$ is $\Omega(pK^{1/3}T^{2/3})$ when $K \leq T$.

Using Yao's principle, for any randomized algorithm π, there exists an instance $\Phi^j_{0,p,\epsilon}$ with $0 \leq j \leq K$ such that $\mathsf{Mod\_Quality\_Reg}_\pi(T, 0, \Phi^j_{0,p,\epsilon}) + \mathsf{Cost\_Reg}_\pi(T, 0, \Phi^j_{0,p,\epsilon})$ is $\Omega(pK^{1/3}T^{2/3})$. Also, since $\mathsf{Mod\_Quality\_Reg}_\pi(T, 0, \Phi^j_{0,p,\epsilon}) \leq \mathsf{Quality\_Reg}_\pi(T, 0, \Phi^j_{0,p,\epsilon})$, we have $\mathsf{Quality\_Reg}_\pi(T, 0, \Phi^j_{0,p,\epsilon}) + \mathsf{Cost\_Reg}_\pi(T, 0, \Phi^j_{0,p,\epsilon})$ is $\Omega(pK^{1/3}T^{2/3})$. ■

Proof of Theorem 2.3.1. **Notation**: For any instance ϕ, we define the arms m^ϕ_* and i^ϕ_* as $m^\phi_* = \arg\max_i \mu^i_\phi$ and $i^\phi_* = \arg\min_i c^i_\phi$ s.t. $q_{i_\phi} \geq (1-\theta)q_{m^*_\phi}$. When the instance is clear, we will use the simplified notation i_* and m_* instead of i^ϕ_* and m^ϕ_*.

Proof Sketch: Lemma 2.3.3 establishes that when $\alpha = 0$, for any given policy, there exists an instance on which the sum of quality and cost regret are $\Omega(K^{1/3}T^{2/3})$. Now, we generalize the above result for $\alpha = 0$ to any α for $0 \leq \alpha \leq 1$. The main idea in our reduction is to show that if there exists an algorithm π_α for $\alpha > 0$ that achieves $o(K^{1/3}T^{2/3})$ regret on every instance in the family $\Phi_{\alpha,p,\epsilon}$, then we can use π_α as a subroutine to construct an algorithm π_0 for problem (2.1) that achieves $o(K^{1/3}T^{2/3})$ regret on every instance in the family $\Phi_{0,p,\epsilon}$, thus contradicting the lower bound of Lemma 2.3.3. This will prove the theorem by contradiction. In order to construct the aforementioned sub-routine, we leverage techniques from *Bernoulli factory* to generate a sample from a Bernoulli random variable with parameter $\mu/(1-\alpha)$ using samples from a Bernoulli random variable with parameter μ, for any $\mu, \alpha < 1$.

Aside on Bernoulli Factory: The key tool we use in constructing the algorithm π_0 from π_α is *Bernoulli factory* for the linear function. The Bernoulli factory for a specified scaling factor $C > 1$ i.e. $BernoulliFactory(C)$ uses a sequence of independent and identically distributed samples from $Ber(r)$ and returns a sample from $Ber(Cr)$.The key aspect of a Bernoulli factory is the number of samples needed from $Ber(r)$ to generate a sample from $Ber(Cr)$. We use the Bernoulli factory described in Huber (2013) which has a guarantee on the expected number of samples τ from $Ber(r)$ needed to generate a sample from $Ber(Cr)$. In particular, for a specified $\delta > 0$,

$$\sup_{r \in [0, \frac{1-\delta}{C}]} E[\tau] \leq \frac{9.5C}{\delta}. \tag{A.3}$$

Detailed proof: For some value of p, ϵ (to be specified later in the proof) such that $0 \leq p < 1$ and $0 \leq \epsilon \leq p/2$, consider the family of instances $\Phi_{\alpha,p,\epsilon}$ and $\Phi_{0,p,\epsilon}$. Let π_α be any algorithm for

the family $\Phi_{\alpha,p,\epsilon}$. Using π_α, we construct an algorithm π_0 for the family $\Phi_{0,p,\epsilon}$. This algorithm is described in Algorithm 6. We will use $I_l^\alpha = \pi_\alpha([(I_1^\alpha, r_1), (I_2^\alpha, r_2), \cdots (I_{l-1}^\alpha, r_{l-1})])$ to denote the arm pulled by algorithm π_α at time l after having observed rewards r_l $\forall 1 \leq i < l$ through arm pulls I_l $\forall 1 \leq i < l$. The function *BernoulliFactory*(C) returns two values - a random sample from the distribution $Ber(Cr)$ and the number of samples of $Ber(r)$ needed to generate this random sample.

Algorithm 6 Derived Algorithm π_0

Require: Algorithm π_α, L - Number of arm pulls for algorithm π_α
  $l = 1, t = 1$
  for $l \in [L]$ **do**
    $I_l^\alpha = \pi_\alpha([(I_1^\alpha, r_1), (I_2^\alpha, r_2), \cdots (I_{l-1}^\alpha, r_{l-1})])$
    if $I_l^\alpha = 0$ **then**
      Pull arm 0 to obtain outcome r_l
      $I_t^0 = I_l^\alpha = 0$
      $U_l = \{t\}$
    else
      Call $r_l, n =$ *BernoulliFactory*$(\frac{1}{1-\alpha})$ on samples generated from repeated pulls of the arm I_l^α
      $U_l = \{t, t+1 \cdots t+n-1\}$
      $I_t^0 = I_{t+1}^0 = \cdots I_{t+n-1}^0 = I_l^\alpha$
    $S_l = |U_l|$
    $l = l + 1$
    $t = t + S_l$
  $T = t$
  return Arm I_t^0 to be pulled in each round t, total number of arm pulls T

Now, let us analyze the expected modified regret incurred by algorithm π_0 on an instance $\Phi_{0,p,\epsilon}^a$ for any $0 \leq a \leq K$ where the expectation is with respect to the random variable T, total number of arm pulls.

Similarly, we analyze the cost regret incurred by algorithm π_0 on an instance $\Phi_{0,p,\epsilon}^a$ for any $0 \leq a \leq K$.

$$
\begin{aligned}
&\mathbb{E}\left[\mathsf{Mod\_Quality\_Reg}_{\pi_0}(T,0,\Phi^a_{0,p,\epsilon})\right] + \mathbb{E}\left[\mathsf{Cost\_Reg}_{\pi_0}(T,0,\Phi^a_{0,p,\epsilon})\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T}\left(\mu_{m^*}^{\Phi^a_{0,p,\epsilon}} - \mu_{I_t^0}^{\Phi^a_{0,p,\epsilon}}\right)\mathbb{1}\{I_t^0=0\}\right] + \mathbb{E}\left[\sum_{t=1}^{T} c_{i_*}^{\Phi^a_{0,p,\epsilon}} - c_{I_t^0}^{\Phi^a_{0,p,\epsilon}}\right] \\
&= \mathbb{E}\left[\sum_{l=1}^{L}\sum_{t\in U_l}\left(\mu_{m^*}^{\Phi^a_{0,p,\epsilon}} - \mu_{I_t^0}^{\Phi^a_{0,p,\epsilon}}\right)\mathbb{1}\{I_t^0=0\}\right] + \mathbb{E}\left[\sum_{l=1}^{L}\sum_{t\in U_l} c_{i_*}^{\Phi^a_{0,p,\epsilon}} - c_{I_t^0}^{\Phi^a_{0,p,\epsilon}}\right] \\
&= \mathbb{E}\left[\sum_{l=1}^{L} S_l\left(\mu_{m^*}^{\Phi^a_{0,p,\epsilon}} - \mu_{I_l^\alpha}^{\Phi^a_{0,p,\epsilon}}\right)\mathbb{1}\{I_l^\alpha=0\}\right] + \mathbb{E}\left[\sum_{l=1}^{L} S_l\left(c_{i_*}^{\Phi^a_{0,p,\epsilon}} - c_{I_l^\alpha}^{\Phi^a_{0,p,\epsilon}}\right)\right] \\
&= \sum_{l=1}^{L}\mathbb{E}\left[\mathbb{E}\left[S_l|I_l^\alpha\right]\left(\mu_{m^*}^{\Phi^a_{0,p,\epsilon}} - \mu_{I_l^\alpha}^{\Phi^a_{0,p,\epsilon}}\right)\mathbb{1}\{I_l^\alpha=0\}\right] + \mathbb{E}\left[\sum_{l=1}^{L}\mathbb{E}[S_l|I_l^\alpha]\left(c_{i_*}^{\Phi^a_{0,p,\epsilon}} - c_{I_l^\alpha}^{\Phi^a_{0,p,\epsilon}}\right)\right] \\
&\le \sum_{l=1}^{L}\mathbb{E}\left[\frac{9.5}{\delta(1-\alpha)}\left(\mu_{m^*}^{\Phi^a_{0,p,\epsilon}} - \mu_{I_l^\alpha}^{\Phi^a_{0,p,\epsilon}}\right)\mathbb{1}\{I_l^\alpha=0\}\right] + \mathbb{E}\left[\sum_{l=1}^{L}\frac{9.5}{\delta(1-\alpha)}\left(c_{i_*}^{\Phi^a_{0,p,\epsilon}} - c_{I_l^\alpha}^{\Phi^a_{0,p,\epsilon}}\right)\right] \\
&= \frac{9.5}{\delta(1-\alpha)}\sum_{l=1}^{L}\mathbb{E}\left[\left((1-\alpha)\mu_{m^*}^{\Phi^a_{\alpha,p,\epsilon}} - \mu_{I_l^\alpha}^{\Phi^a_{0,p,\epsilon}}\right)\mathbb{1}\{I_l^\alpha=0\}\right] + \frac{9.5}{\delta(1-\alpha)}\sum_{l=1}^{L}\mathbb{E}\left[c_{i_*}^{\Phi^a_{\alpha,p,\epsilon}} - c_{I_l^\alpha}^{\Phi^a_{\alpha,p,\epsilon}}\right] \\
&= \frac{9.5}{\delta(1-\alpha)}\mathsf{Quality\_Reg}_{\pi_\alpha}(L,\alpha,\Phi^a_{\alpha,p,\epsilon}) + \frac{9.5}{\delta(1-\alpha)}\mathsf{Cost\_Reg}_{\pi_\alpha}(L,\alpha,\Phi^a_{\alpha,p,\epsilon}).
\end{aligned}
$$

The third last line is based on using (A.3) and the second last line holds because costs of arms are same in all instances, $i_*^{\Phi^a_{\alpha,p,\epsilon}} = i_*^{\Phi^a_{0,p,\epsilon}} = a$ and $\mu_{m^*}^{\Phi^a_{0,p,\epsilon}} = (1-\alpha)\mu_{m^*}^{\Phi^a_{\alpha,p,\epsilon}}$.

Thus,

$$
\begin{aligned}
&\mathsf{Quality\_Reg}_{\pi_\alpha}(L,\alpha,\Phi^a_{\alpha,p,\epsilon}) + \mathsf{Cost\_Reg}_{\pi_\alpha}(L,\alpha,\Phi^a_{\alpha,p,\epsilon}) \\
&\ge \frac{\delta(1-\alpha)}{9.5}\mathbb{E}\left[\mathsf{Mod\_Quality\_Reg}_{\pi_0}(T,0,\Phi^a_{0,p,\epsilon}) + \mathsf{Cost\_Reg}_{\pi_0}(T,0,\Phi^a_{0,p,\epsilon})\right] \qquad \text{(A.4)} \\
&\ge \frac{\delta(1-\alpha)}{9.5}\mathbb{E}\left[\mathsf{Mod\_Quality\_Reg}_{\pi_0}(L,0,\Phi^a_{0,p,\epsilon}) + \mathsf{Cost\_Reg}_{\pi_0}(L,0,\Phi^a_{0,p,\epsilon})\right] \qquad \text{(because } L\le T\text{)} \\
&\ge \frac{\delta(1-\alpha)}{9.5}\left(\mathsf{Mod\_Quality\_Reg}_{\pi_0}(L,0,\Phi^a_{0,p,\epsilon}) + \mathsf{Cost\_Reg}_{\pi_0}(L,0,\Phi^a_{0,p,\epsilon})\right) \qquad \text{(A.5)}
\end{aligned}
$$

Using Lemma 2.3.3 and choosing $p=\frac{1-\alpha}{3}, \delta=\frac{1}{2}, \epsilon=\frac{p}{2}(\frac{K}{T})^{1/3}$, we get for any randomized algorithm π_α, there exists instance $\Phi^b_{\alpha,p,\epsilon}$ (for some $0\le b\le K$) such that $\mathsf{Quality\_Reg}_{\pi}(T,\alpha,\Phi^b_{\alpha,p,\epsilon}) + \mathsf{Cost\_Reg}_{\pi}(T,\alpha,\Phi^b_{\alpha,p,\epsilon})$ is $\Omega\left((1-\alpha)^2K^{1/3}T^{2/3}\right)$.

■

A.3. Performance of Algorithms

We use the following fact in the proof of Theorem 2.2.1.

Fact 1. *(Abramowitz and Stegun 1948) For a Normal random variable Z with mean m and variance σ^2, for any z,*

$$P\left(|Z-m|>z\sigma\right) > \frac{1}{4\sqrt{\pi}}\exp(-\frac{7z^2}{2}).$$

Proof of Theorem 2.2.1. This proof is inspired by the lower bound proof in Agrawal and Goyal (2017b). For any given α, K and T, we construct an instance on which the CS-TS algorithm (Algorithm 1) gives linear regret in cost.

Consider an instance ϕ with K arms where the costs and mean reward of the j-th arm are

$$c_j = \begin{cases} 0 & j=0 \\ 1 & j\neq 0 \end{cases}, \qquad \mu_j = \begin{cases} (1-\alpha)q + \frac{d}{\sqrt{T}} & j=0 \\ q & j\neq 0 \end{cases}$$

where $q = \frac{d}{(1-\alpha)\sqrt{T}}$ for some $0 < d < \min\{\sqrt{T}/2, (1-\alpha)\sqrt{T}\}$. Moreover, the reward of each arm is deterministic though this fact is not known to the agent. As in the SMS application, we assume that the cost rewards of all arms are known a priori to the agent.

Let the prior distribution that the agent assumes over the mean reward of each arm be $\mathcal{N}(0,\sigma_0^2)$ for some prior variance σ_0^2 . Further, the agent assumes that the observed qualities to be normally distributed with noise variance σ_n^2. As such at the start of period t, the agent will consider a normal posterior distribution for each arm i with mean

$$\hat{\mu}_i(t) = \frac{T_i(t)}{\frac{\sigma_n^2}{\sigma_0^2} + T_i(t)}\mu_i \tag{A.6}$$

and variance

$$\sigma_i(t)^2 = \left(\frac{1}{\sigma_0^2} + \frac{T_i(t)}{\sigma_n^2}\right)^{-1}. \tag{A.7}$$

As $d < q\alpha\sqrt{T}$, the highest quality across all arms is q. Thus, note that all arms are *feasible* in terms of quality i.e. have their quality within $(1-\alpha)$ factor of the best quality arm. Hence, quality regret $\mathsf{Quality\_Reg}_{\mathsf{CS-TS}}(t,\alpha,\phi) = 0\ \forall t > 0$ (for any algorithm) on this instance.

The first arm is the optimal arm (i_*). Thus, the cost regret equals the number of times any arm but the first arm is pulled. In particular, let

$$R_c(T) = \sum_{t=1}^{T} \max\{c_{I_t} - c_{i_*}, 0\} = \sum_{t=1}^{T} \mathbf{1}\{I_t \neq 1\},$$

so that $\mathsf{Cost\_Reg}_{\mathsf{CS-TS}}(T,\alpha,I) = \mathbb{E}\left[R_C(T)\right]$.

Define the event $A_{t-1} = \{\sum_{i\neq 1} T_i(t) \leq sT\sqrt{K}\}$ for a fixed constant $s > 0$. For any t, if the event A_{t-1} is not true, then $R_c(T) \geq R_c(t) \geq sT\sqrt{K}$. We can assume that $P(A_{t-1}) \geq 0.5\ \forall t \leq T$.

Otherwise

$$\begin{aligned}\mathsf{Cost\_Reg}_{\mathsf{CS-TS}}(T,\alpha,\phi) &= \mathbb{E}\left[R_C(T)\right] \\ &\geq 0.5E[R_C(T)|A_{t-1}^c] \\ &= \Omega(T\sqrt{K}).\end{aligned}$$

Now, we will show that whenever A_{t-1} is true, probability of playing a sub-optimal arm is at least a constant. For this, we show that the probability that $\mu_1^{score}(t) \leq \mu_1$ and $\mu_i^{score}(t) \geq \frac{\mu_1}{1-\alpha}$, for some $1 < i \leq K$ is lower bounded by a constant.

Now, given any history of arm pulls $\mathcal{F}_{t-1}$ before time t, $\mu_1^{score}(t)$ is a Gaussian random variable with mean $\hat{\mu}_1(t) = \frac{T_i(t)}{\frac{\sigma_n^2}{\sigma_0^2}+T_i(t)}\mu_1$. By symmetry of Gaussian random variables, we have

$$\begin{aligned}P\left(\mu_1^{score}(t) \leq \mu_1 \middle| \mathcal{F}_{t-1}\right) &\geq P\left(\mu_1^{score}(t) \leq \frac{T_i(t)}{\frac{\sigma_n^2}{\sigma_0^2}+T_i(t)}\mu_1 \middle| \mathcal{F}_{t-1}\right) \\ &= P\left(\mu_1^{score}(t) \leq \hat{\mu}_1(t) \middle| \mathcal{F}_{t-1}\right) \\ &= 0.5.\end{aligned}$$

Based on (A.6) and (A.7), given any realization F_{t-1} of $\mathcal{F}_{t-1}$, $\mu_i^{score}(t)$ for $i \neq 1$ are independent Gaussian random variables with mean $\hat{\mu}_i(t)$ and variance $\sigma_i(t)^2$. Thus, we have

$$
\begin{aligned}
&P\left(\exists i \neq 1,\ \mu_i^{score}(t) \geq \frac{\mu_1}{1-\alpha} \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right) \\
&= P\left(\exists i \neq 1,\ \mu_i^{score}(t) - \hat{\mu}_i(t) \geq \frac{1}{1-\alpha}\left(q(1-\alpha) + \frac{d}{\sqrt{T}}\right) - \hat{\mu}_i(t) \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right) \\
&= P\left(\exists i \neq 1,\ \mu_i^{score}(t) - \hat{\mu}_i(t) \geq \frac{d}{(1-\alpha)\sqrt{T}} + \frac{1}{1+T_i(t)\frac{\sigma_0^2}{\sigma_n^2}} q \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right) \\
&\geq P\left(\exists i \neq 1,\ \mu_i^{score}(t) - \hat{\mu}_i(t) \geq \frac{d}{(1-\alpha)\sqrt{T}} + q \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right) \\
&= P\left(\exists i \neq 1,\ \mu_i^{score}(t) - \hat{\mu}_i(t) \geq \frac{2d}{(1-\alpha)\sqrt{T}} \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right) \\
&= P\left(\exists i \neq 1,\ (\mu_i^{score}(t) - \hat{\mu}_i(t))\frac{1}{\sigma_i(t)} \geq \left(\frac{2d}{(1-\alpha)\sqrt{T}}\right)\frac{1}{\sigma_i(t)} \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right) \\
&= P\left(\exists i \neq 1,\ Z_i(t) \geq \left(\frac{2d}{(1-\alpha)\sqrt{T}}\right)\frac{1}{\sigma_i(t)} \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right)
\end{aligned}
$$

where $Z_i(t)$ are independent standard normal variables for all i, t. Thus,

$$
\begin{aligned}
&P\left(\exists i \neq 1,\ \mu_i^{score}(t) \geq \frac{\mu_1}{1-\alpha} \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right) \\
&= 1 - P\left(\forall i \neq 1,\ Z_i(t) \leq \left(\frac{2d}{(1-\alpha)\sqrt{T}}\right)\frac{1}{\sigma_i(t)} \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right) \\
&= 1 - \Pi_{i\neq 1}\left(1 - P\left(\ Z_i(t) \geq \left(\frac{2d}{(1-\alpha)\sqrt{T}}\right)\frac{1}{\sigma_i(t)} \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right)\right) \\
&\geq 1 - \Pi_{i\neq 1}\left(1 - \frac{1}{8\sqrt{\pi}}\exp\left(-\frac{7}{2}\frac{1}{\sigma_i(t)^2}\left(\frac{2d}{(1-\alpha)\sqrt{T}}\right)^2\right)\right) \qquad \text{(Using Fact 1)} \\
&= 1 - \Pi_{i\neq 1}\left(1 - \frac{1}{8\sqrt{\pi}}\exp\left(-\frac{7}{2}\left(\frac{1}{\sigma_0^2} + \frac{T_i(t)}{\sigma_n^2}\right)\left(\frac{2d}{(1-\alpha)\sqrt{T}}\right)^2\right)\right) \\
&\geq 1 - \Pi_{i\neq 1}\left(1 - \frac{1}{8\sqrt{\pi}}\exp\left(-\frac{7}{2}\left(\frac{1}{\sigma_0^2} + \frac{1}{\sigma_n^2}\right)T_i(t)\left(\frac{2d}{(1-\alpha)\sqrt{T}}\right)^2\right)\right),
\end{aligned}
$$

The last inequality follows from the fact that $T_i(t) \geq 1$.

Now, when the event A_{t-1} holds, we have $\sum_{i\neq 1} T_i(t) \leq sT\sqrt{K}$. Thus, the right hand side would be minimized when $T_i(t) = \frac{sT}{\sqrt{K}}$, $\forall i \neq 1$. Substituting this value of $T_i(t)$, the right hand side reduces to $g(K) = 1 - \Pi_{i\neq 1}\left(1 - \frac{1}{8\sqrt{\pi}}\exp\left(-14\left(\frac{1}{\sigma_0^2} + \frac{1}{\sigma_n^2}\right)\frac{s\sqrt{K}d^2}{(1-\alpha)^2}\right)\right)$. Thus, $P\left(\exists i \neq 1,\ \mu_i^{score}(t) \geq \frac{\mu_1}{1-\alpha} \,\middle|\, \mathcal{F}_{t-1} = F_{t-1}\right) \geq g(K)$ whenever F_{t-1} is such that A_{t-1} holds.

Probability of playing any sub-optimal arm at time t is,

$$\begin{aligned}
P\left(\exists i \neq 1,\ I_t = i\right) &\geq P\left(\mu_1^{score}(t) \leq \mu_1,\ \exists i \neq 1 \text{ s.t. } \mu_i^{score}(t) \geq \frac{\mu_1}{1-\alpha}\right) \\
&= \mathbb{E}\left[P\left(\mu_1^{score}(t) \leq \mu_1,\ \exists i \neq 1 \text{ s.t. } \mu_i^{score}(t) \geq \frac{\mu_1}{1-\alpha}\right)\middle|\mathcal{F}_{t-1}\right] \\
&\geq \mathbb{E}\left[P\left(\mu_1^{score}(t) \leq \mu_1,\ \exists i \neq 1 \text{ s.t. } \mu_i^{score}(t) \geq \frac{\mu_1}{1-\alpha}\right)\middle|\mathcal{F}_{t-1}, A_{t-1}\right] P(A_{t-1}) \\
&\geq \frac{1}{2} \cdot g(K) \cdot \frac{1}{2}
\end{aligned}$$

Thus, at every time instant t, the probability of playing a sub-optimal is lower bounded by $\frac{g(K)}{4}$. This implies that the cost regret $\mathsf{Cost\_Reg}_{\mathsf{CS-TS}}(T, \alpha, \phi) \geq 0.25Tg(K)$.

■

Proof of Theorem 2.4.1. This algorithm has two phases - pure exploration and UCB. In the first phase, the algorithm pulls each arm a specified number of times (τ). In the second phase, the algorithms maintains upper and lower confidence bounds on the mean reward of each arm. Then, it estimates a *feasible* set of arms and pulls the cheapest arm in this set.

We will define the *clean event* $\mathcal{E}$ in this proof as the event that for every time $t \in [T]$ and arm $i \in [K]$, the difference between the mean reward and the empirical mean reward does not exceed the size of the confidence interval ($\beta_i(t)$) i.e. $\mathcal{E} = \{|\hat{\mu}_i(t) - \mu_i| \leq \beta_i(t),\ \forall i \in [K],\ t \in [T]\}$.

Define $\hat{t} = K\tau + 1$ as the first round in the UCB phase of the algorithm. Further, define instantaneous cost and quality regret as the regret incurred in the t-th arm pull:

$$\begin{aligned}
\mathsf{Quality_Reg}_{\pi}^{inst}(t, T, \alpha, \boldsymbol{\mu}, \mathbf{c}) &= \mathbb{E}\left[\max\{(1-\alpha)\mu_{m_*} - \mu_{\pi_t}, 0\}\right], \\
\mathsf{Cost_Reg}_{\pi}^{inst}(t, T, \alpha, \boldsymbol{\mu}, \mathbf{c}) &= \mathbb{E}\left[\max\{c_{\pi_t} - c_{i_*}, 0\}\right],
\end{aligned} \tag{A.8}$$

where the expectation is over the randomness in the policy π.

Let us first assume that the clean event holds. As both the instantaneous regrets are upper bounded by 1, $\sum_{t=1}^{K\tau} \mathsf{Quality\_Reg}_{\pi}^{inst}(t, T, \alpha, \boldsymbol{\mu}, \mathbf{c}) \leq K\tau$ and $\sum_{t=1}^{K\tau} \mathsf{Cost\_Reg}_{\pi}^{inst}(t, T, \alpha, \boldsymbol{\mu}, \mathbf{c}) \leq K\tau$.

Now, let us look at the UCB phase of the algorithm. Here, $\forall\ \hat{t} \leq t \leq T$, we have

$$\mu_{i_*}^{\mathsf{UCB}}(t) \geq \mu_{i_*} \geq (1-\alpha)\mu_{m_*} \geq (1-\alpha)\mu_{m_t} \geq (1-\alpha)\mu_{m_t}^{\mathsf{LCB}}(t).$$

Here, the first and fourth inequality are because of the clean event. The second and third

inequality are from the definition of i_* and m_* respectively.

Thus from the inequality above, the optimal arm i_* is in the set $Feas(t), \forall \hat{t} \leq t \leq T$. This implies that the arm pulled in each time step in the UCB phase, is either the optimal arm or an arm cheaper than it. Thus, instantaneous cost regret is zero for all time steps in the UCB phase of the algorithm.

Now, let us look at the quality regret in the UCB phase i.e. for any $\hat{t} \leq t \leq T$. We have $\mu_{I_t} + 2\beta_{I_t}(t) \geq \mu_{I_t}^{\mathsf{UCB}}(t) \geq (1-\alpha)\mu_{m_t}^{\mathsf{LCB}}(t) \geq (1-\alpha)\mu_{m_*}^{\mathsf{LCB}}(t) \geq (1-\alpha)\left(\mu_{m_*} - 2\beta_{m_*}(t)\right) \geq (1-\alpha)\mu_{m_*} - 2\beta_{m_*}(t)$. The first and fourth inequality hold because the clean event holds. The second and third inequalities follow from the definition of I_t and m_t respectively. Thus,

$$\begin{aligned}
\mathsf{Quality_Reg}_{\pi}^{inst}(t, T, \alpha, \boldsymbol{\mu}, \mathbf{c}|\mathcal{E}) &= (1-\alpha)\mu_{m_*} - \mu_{I_t} \\
&\leq 2\left(\beta_{I_t}(t) + \beta_{m_*}(t)\right) \\
&\leq 2\left(\sqrt{\frac{2\log T}{\tau}} + \sqrt{\frac{2\log T}{\tau}}\right) \\
&= 4\sqrt{\frac{2\log T}{\tau}}.
\end{aligned}$$

The total regret incurred by the algorithm is the sum of the instantaneous regrets across all time steps in the exploration and the UCB phase. Thus,

$\mathsf{Quality\_Reg}_{\pi}(T, \alpha, \boldsymbol{\mu}, \mathbf{c}|\mathcal{E}) \leq K\tau + 4(T - K\tau)\sqrt{\frac{2\log T}{\tau}} \leq K\tau + 4T\sqrt{\frac{2\log T}{\tau}}$ and $\mathsf{Cost\_Reg}_{\pi}(T, \alpha, \boldsymbol{\mu}, \mathbf{c}|\mathcal{E}) \leq K\tau$. Substituting $\tau = (T/K)^{2/3}$, we conclude that both cost and quality regret are

$O(K^{1/3}T^{2/3}\sqrt{\log T})$.

Now, when the clean event does not hold, the cost and quality regret are at most T each. The probability that the clean event does not hold is at most $2/T^2$ (Lemma 1.6 in Slivkins (2019)). Thus, the expected cost and quality regret obtained by averaging over the clean event holding and not holding is $O(K^{1/3}T^{2/3}\sqrt{\log T})$.

■

Proof of Theorem 2.5.1. As in the previous proof, we will define the *clean event* $\mathcal{E}$ as the event that for every time $t \in [T]$ and arm $i \in [K]$, the difference between the mean reward and the empirical mean reward does not exceed the size of the confidence interval $(\beta_i(t))$ i.e. $\mathcal{E} = \{|\hat{\mu}_i(t) - \mu_i| \leq \beta_i(t),\ \forall i \in [K],\ t \in [T]\}$. Also, define the quality and cost gap of each arm

as $\Delta_{\mu,i} = \max\{(1-\alpha)\mu_{m^*} - \mu_i, 0\}$ and $\Delta_{c,i} = \max\{c_{i_*} - c_i, 0\}$.

When the clean event does not hold, both cost and quality regrets are upper bounded by T. Let us look at the case when the clean event holds and analyze the cost and quality regret.

Quality Regret: Let t_i be the last time t when $i \in Feas(t)$ i.e. $t_i = \max\{K,\ \max\{t : i \in Feas(t)\}\}$. Thus, $T_i(T) = T_i(t_i)$.

Consider any arm i which would incur a quality regret on being pulled i.e. arm i such that $\mu_i < (1-\alpha)\mu_{m_*}$. We have

$$\mu_i + 2\beta_i(t_i) \geq \mu_i^{\mathsf{UCB}}(t_i) \geq (1-\alpha)\mu_{m_{t_i}}^{\mathsf{UCB}}(t_i) \geq (1-\alpha)\mu_{m_*}^{\mathsf{UCB}}(t_i) \geq (1-\alpha)\mu_{m_*}.$$

The first and fourth inequality hold because of the clean event. The third inequality is from the definition of m_{t_i}.

Thus, $(1-\alpha)\mu_{m_*} - \mu_i \leq 2\beta_i(t_i)$. Using the definition of $\beta_i(t_i)$, we get $T_i(T) = T_i(t_i) \leq \frac{8\log T}{\Delta_{\mu,i}^2}$.

Using Jensen's inequality,

$$\begin{aligned}
\left(\frac{\sum_{i=1}^K T_i(T)\Delta_{\mu,i}}{T}\right)^2 &\leq \frac{\sum_{i=1}^K T_i(T)\Delta_{\mu,i}^2}{T} \\
&= \frac{\sum_{i=1:\ \Delta_{\mu,i}>0}^K T_i(T)\Delta_{\mu,i}^2}{T} \\
&\leq \sum_{i=1:\Delta_{\mu,i}>0}^{K} \frac{8\log T}{\Delta_{\mu,i}^2}\frac{\Delta_{\mu,i}^2}{T} \\
&= \frac{8K\log T}{T}
\end{aligned}$$

Thus, $\mathsf{Quality\_Reg}_\pi(T, \alpha, \boldsymbol{\mu}, \mathbf{c}|\mathcal{E}) \leq \sqrt{8KT\log T}$.

Cost Regret: Let i be an arm such that $c_i > c_{i_*}$. Let $\tilde{t}_i$ be the last time when arm i is pulled. Thus, $i_* \notin Feas(\tilde{t}_i)$. We have, $\mu_{i_*} \leq \mu_{i_*}^{\mathsf{UCB}}(\tilde{t}_i) < \mu_i^{\mathsf{UCB}}(\tilde{t}_i) \leq \mu_i(\tilde{t}_i) + 2\sqrt{(2\log T)/T_i(\tilde{t}_i)}$. Thus,

$$T_i(T) = T_i(\tilde{t}_i) < \frac{8\log T}{(\mu_{i_*} - \mu_i)^2} \leq \frac{8\delta^2\log T}{(c_{i_*} - c_i)^2} = \frac{8\delta^2\log T}{\Delta_{c,i}^2}.$$

Using Jensen's inequality as for the case of quality regret, we get, $\mathsf{Cost\_Reg}_\pi(T, \alpha, \boldsymbol{\mu}, \mathbf{c}|\mathcal{E}) \leq \sqrt{8\delta^2 KT\log T}$.

Note that the probability of the clean event is at least $1 - 2/T^2$ (Lemma 1.6 in Slivkins (2019)). Thus, the sum of the expected cost and quality regret by averaging over the clean event holding and not holding is $O((1+\delta)\sqrt{KT\log T})$.

■

A.4. Algorithm with Unknown and Random Costs

Algorithm 7 CS-ETC with unknown costs

Require: K, T, Number of exploration pulls τ
  $T_i(1) = 0 \ \forall i \in [K]$
  Pure exploration phase:
  for $t \in [1, Kf(K,T)]$ **do**
    $I_t = t \mod K$
    Pull arm I_t to obtain reward r_t and cost χ_t
    $T_i(t+1) = T_i(t) + \mathbf{1}\{I_t = i\} \ \forall i \in [K]$
  UCB phase:
  for $t \in [Kf(K,T)+1, T]$ **do**
    $\hat{\mu}_i(t) \leftarrow \frac{\sum_{\tau=1}^{t-1} r_\tau \mathbb{1}\{I_\tau = i\}}{T_i(t)} \ \forall i \in [K]$
    $\hat{c}_i(t) \leftarrow \frac{\sum_{\tau=1}^{t-1} \chi_\tau \mathbb{1}\{I_\tau = i\}}{T_i(t)} \ \forall i \in [K]$
    $\beta_i(t) \leftarrow \sqrt{\frac{2 \log T}{T_i(t)}} \ \forall i \in [K]$
    $\mu_i^{\mathsf{UCB}}(t) \leftarrow \min\{\hat{\mu}_i(t) + \beta_i(t), 1\} \ \forall i \in [K]$
    $\mu_i^{\mathsf{LCB}}(t) \leftarrow \max\{\hat{\mu}_i(t) - \beta_i(t), 0\} \ \forall i \in [K]$
    $c_i^{\mathsf{LCB}}(t) \leftarrow \max\{\hat{c}_i(t) - \beta_i(t), 0\} \ \forall i \in [K]$
    $m_t = \arg\max_i \mu_i^{\mathsf{LCB}}(t)$
    $Feas(t) = \{i : \mu_i^{\mathsf{UCB}}(t) > (1-\alpha)\mu_{m_t}^{\mathsf{LCB}}(t)\}$
    $I_t = \arg\min_{i \in Feas(t)} c_i^{\mathsf{LCB}}$
    Pull arm I_t to obtain reward r_t and cost χ_t
    $T_i(t+1) = T_i(t) + \mathbf{1}\{I_t = i\} \ \forall i \in [K]$
  return Arm I_t to be pulled in each round $t \in [T]$

Appendix B

Appendix to Chapter 3

B.1. Sample Complexity of the Sample Average Approximation

Lemma B.1.1. Let $U_1, \ldots, U_m$ be i.i.d. samples from distribution U, and let S_m^* be an optimal solution to

$$\max_{S \subset V, |S| \leq k} \frac{1}{m} \sum_{i \in [m]} f(S, U_i).$$

There exists some universal constant C such that for any $\epsilon \in (0, 1]$,

$$\mathsf{E}[f(S_m^*, U)] \geq \mathrm{OPT} - \epsilon$$

as long as

$$m \geq C \frac{k \log n}{\epsilon^2}.$$

Proof of Lemma B.1.1. This follows from a standard tail bound for bounded (or sub-gaussian, more generally) variables. Fix any $t > 0$. For any $S \subset V$, the random variable $f(S, U)$ lies in the interval $[0, 1]$ by assumption, and so by Hoeffding's inequality,

$$\mathsf{P}\left(\left| \frac{1}{m} \sum_{i \in [m]} f(S, U_i) - \mathsf{E}[f(S, U)] \right| > t \right) \leq 2 \exp\left(-cmt^2\right),$$

for some universal constant c. Applying a union bound over all cardinality-constrained $S \subset V$,

we obtain:

$$\begin{aligned}
&\mathsf{P}\left(\max_{S\subset V,|S|\le k}\left|\frac{1}{m}\sum_{i\in[m]} f(S,U_i) - \mathsf{E}[f(S,U)]\right| > t\right) \\
&\le \sum_{S\subset V,|S|\le k} \mathsf{P}\left(\left|\frac{1}{m}\sum_{i\in[m]} f(S,U_i) - \mathsf{E}[f(S,U)]\right| > t\right) \\
&\le 2kn^k \exp\left(-cmt^2\right),
\end{aligned}$$

which implies (e.g. by integrating over $t \ge 0$) that for some constant C,

$$\mathsf{E}\left[\max_{S\subset V,|S|\le k}\left|\frac{1}{m}\sum_{i\in[m]} f(S,U_i) - \mathsf{E}[f(S,U)]\right|\right] \le C\sqrt{\frac{k\log n}{m}}.$$

The error incurred by optimizing over the sample average approximation is at most twice this uniform bound, which equals ϵ for m as given in the statement of the theorem. ■

B.2. Additional Proofs

B.2.1. Proof of Lemma 3.4.2

First, fix any $u \in \mathcal{M}$ and $v \in V$, and let $r_0 = \lceil -\log_2 p(d(v,u))\rceil$. If $p(d(v,u)) \le 2\rho_0$, then clearly

$$\mathsf{P}(v \in \mathrm{LSS}[V,p,c](u)) \ge \ \rho_0 \ge \ p(d(v,u))/2.$$

Now, let us consider the case when $p(d(v,u)) > 2\rho_0$. Here we have

$$\begin{aligned}
\mathsf{P}(v \in \mathrm{LSS}[V,p,c](u)) &\ge \mathsf{P}\left(v \in \bigcup_{r=r_0}^{R} \mathrm{ANN}[\rho_r V, \gamma_r, c, 1/2](u)\right) \\
&= 1 - \prod_{r=r_0}^{R} \mathsf{P}\left(v \notin \mathrm{ANN}[\rho_r V, \gamma_r, c, 1/2](u)\right) \\
&\ge 1 - \prod_{r=r_0}^{R} (1 - \rho_r/2) \\
&\ge \frac{1/2}{2^{r-1}} \\
&\ge p(d(v,u))/2,
\end{aligned}$$

which is precisely the first statement.

Now, by Assumption 3.2.2, we can guarantee $O(Rn^\alpha)$ runtime as long as the expected number

of items returned is $O(n^\beta)$. This is indeed the case:

$$\begin{aligned}\sum_{r=1}^{R} \rho_r \sum_{v \in V} \mathbb{1}\{d(v_j, u) \leq c\gamma_r\} &= \sum_{v \in V} \sum_{r=1}^{R} \rho_r \mathbb{1}\{d(v_j, u) \leq c\gamma_r\} \\ &\leq \sum_{v \in V} \max\left\{2p\left(\frac{d(v,u)}{c}\right), \frac{1}{n}\right\} \\ &\leq 2n^\beta\end{aligned}$$

■

B.2.2. Proof of Proposition 3.4.3

Two facts follow from the choice of parameters in the LSH data structures. First, for any v such that $p(u, v) \in (\rho_r/2, \rho_r]$, the probability that the algorithm returns v is at most ρ_r and at least $\rho_r[1 - (1 - q(\gamma_r)^{a_r})^{b_r}]$. Since we have

$$\begin{aligned}(1 - q(\gamma_r)^{a_r})^{b_r} &\leq \exp\left(-q(\gamma_r)^{a_r} b_r\right) \\ &\leq \exp\left(-q(\gamma_r)(2^r n^{\beta-1})^{\log_{q(c\gamma_r)} q(\gamma_r)} \log(2) 2^{-r\delta} n^{\delta(1-\beta)} (1/q(\gamma_r))\right) \\ &\leq 1/2,\end{aligned}$$

it follows that this satisfies the sampling requirement.

Second, the expected total number of collisions in a given LSH structure is at most

$$\begin{aligned}(2n^\beta + \rho_r n q(c\gamma_r)^{a_r}) b_r &\leq (2n^\beta + \rho_r n 2^r n^{\beta-1}) b_r \\ &= 4n^\beta b_r \\ &\leq 2^{2-r\delta} n^{\beta+\delta(1-\beta)} (1/q(\gamma_r)) \log(2) + 4n^\beta.\end{aligned}$$

The first inequality follows from the definition of a_r, which implies that

$$a_r \geq \log_{q(c\gamma_r)} 2^r n^{\beta-1}.$$

The equality comes from the definition of ρ_r and combining terms. The second inequality follows from the definition of b_r, which implies that

$$b_r \leq \log(2) 2^{-r\delta} n^{\delta(1-\beta)} (1/q(\gamma_r)) + 1.$$

Thus, across all structure the expected number of collisions is $O(n^{\beta+\delta(1-\beta)} \log n)$.

Appendix C

Appendix to Chapter 4

C.1. Data Augmentation before MLE for the BundleMVL-K Model

If we want to estimate a BundleMVL-K model where K is smaller than the size of some purchased bundles in the dataset, then we can pre-process these observations. In particular, we first partition the purchased bundle, which is of size larger than K, into subsets of size at most K, and augment each such subset as an additional observation. We consider all possible such partitions and assign them equal probability weights (see Algorithm 8). The MLE objective is also updated to incorporate these importance weights.

Algorithm 8 Dataset Pre-processing

Require: Purchased bundles $\tilde{S}_1, \cdots \tilde{S}_{\tilde{m}}$.
  $\mathcal{S} \leftarrow [\,]$, weights $\leftarrow [\,]$, $l \leftarrow 1$.
  while $l \leq \tilde{m}$ **do**
    if $|\tilde{S}_l| \leq K$ **then**
      $\mathcal{S}$.append($\tilde{S}_l$).
      weights.append(1).
    else
      Let $Q_l = \{A = (A_1, \cdots A_t) : \exists\ l$ s.t. $\cup_{j=1}^{t} A_j = \tilde{S}_i, |A_j| \leq K\ \forall\ 1 \leq j \leq t$; and $A_1, A_2, \cdots A_t$ are pairwise disjoint$\}$.
      for R in Q_l **do**
        $\mathcal{S}$.append(A).
        weights.append($\frac{1}{|Q_l|}$).
    $l \leftarrow l + 1$.
  return $\mathcal{S}$, weights

Algorithm 9 MIP Formulation (BundleMVL-2 model)

$$
\begin{aligned}
\max\ & \sum_{i \in W} \sum_{j \in W} \hat{r}_{ij} p_{ij} \\
\text{s.t. } & p_{ij} \leq x_{ij}\ \forall\ i, j \in W, \\
& p_{ij} \leq \frac{V_{\{i,j\}}}{v_0} p_{00}\ \forall\ i, j \in W \\
& p_{ij} + \frac{V_{\{i,j\}}}{v_0}(1 - x_{ij}) \geq \frac{V_{\{i,j\}}}{v_0} p_{00}\ \forall\ i, j \in W \\
& x_{ii} + x_{jj} - 1 \leq x_{ij}\ \forall\ i, j \in W \\
& x_{ij} \leq \min(x_{ii}, x_{jj})\ \forall\ i, j \in W \\
& p_{00} + \sum_{i \in W} \sum_{j \in W} p_{ij} = 1 \\
& x_{ij} \in \{0, 1\}\ \forall\ i, j\ \in W \\
& p_{ij} \geq 0\ \forall\ i, j \in W \\
& p_{00} \geq 0.
\end{aligned}
$$

C.2. Unconstrained Revenue Optimization under the BundleMVL-2 Model: Hardness Result and Structural Properties of the Optimal Solution

Proof of Theorem 4.3.1. Consider the decision version of the unconstrained BundleMVL-2 optimization problem:

$$\max_{C\in 2^W} R_2(C) \geq \kappa \iff \max_{C\in 2^W} \sum_{i\in W}\sum_{j\in W} \theta_{ij} x_i^C x_j^C (\hat{r}_{ij} - \kappa) \geq \kappa v_0 \qquad \text{(Compare-Step)}$$

We will show that this decision version of the revenue optimization problem under the BundleMVL-2 model is NP-complete by a reduction from Max-Cut to this problem. Without loss of generality, we can assume that revenues of all products is less than κ (if not, then these products will be in the recommendation set corresponding to the solution of the optimization problem). Consider a graph G with nodes $\{1, \cdots m\}$. We obtain a modified graph G' by removing all the self-loops in G. Let the adjacency matrix of G' be A'. Let $\mathbf{d} = (d_1, \cdots d_m)$ denote a m-dimensional vector with the i-th entry being the degree of node i in the graph G'. Consider the following $(m+1)\times(m+1)$ matrix $Q = \left(\begin{smallmatrix} 0 & \mathbf{d/2} \\ \mathbf{d/2} & -A' \end{smallmatrix}\right)$ with a generic entry $q_{i,j}$ (in the i-th row and the j-th column). We index the entries of this matrix starting from 0 and the nodes of the graph starting from 1. Hence, for $i > 0$, the i-th column of the Q matrix corresponds to the i-th node in the graph G'. Consider the optimization problem:

$$\arg\max_{C\in 2^W} \sum_{0\leq i\leq m}\sum_{0\leq j\leq m} q_{i,j} x_i^C x_j^C, \tag{C.1}$$

with solution C^*. This optimization problem is equivalent to the Max-Cut problem on the graph G as shown below. Note that the only positive values $q_{i,j}$ are either in the first row or the first column, hence $x_0^{C^*} = 1$. Now,

$$\begin{aligned}
\sum_{0\leq i\leq m}\sum_{0\leq j\leq m} q_{i,j} x_i^{C^*} x_j^{C^*} &= 2\sum_{1\leq j\leq m} q_{0,j} x_j^{C^*} + 2\sum_{1\leq i\leq m}\sum_{1\leq j\leq m,\ j>i} q_{i,j} x_i^{C^*} x_j^{C^*} \\
&= 2\sum_{j\in C^*} \frac{d_j}{2} - 2E_{G'}\left(\tilde{C}^*, \tilde{C}^*\right) \\
&= E_G\left(\tilde{C}^*, \{1,\cdots m\}\backslash\{\tilde{C}^*\}\right),
\end{aligned}$$

where $E_{G'}(C, C')$ represents the number of edges between the set of nodes C and C' in the graph G', and $\tilde{C}^* = C^*\backslash\{0\}$. The final expression equals the number of edges across the cut

$\left(\tilde{C}^*, \{1, \cdots m\} \backslash \{\tilde{C}^*\}\right)$ in the graph G.

We can also see that problem (C.1) can be transformed into an equivalent COMPARE-STEP problem as follows: choose numbers $\kappa, r_0, r_1, \cdots r_m$ such that $r_0 > \kappa$ and $\kappa/2 > r_1 > r_2 \cdots r_m > 0$. Let $\theta_{ij} = \frac{q_{ij}}{r_i + r_j - \kappa}, \quad 0 \leq i, j \leq m$. Thus, the problem of finding the maximum cut on any graph can be transformed to the optimization problem in the COMPARE-STEP of a BundleMVL-2 optimization problem. Moreover, given a solution of the COMPARE-STEP optimization problem, the maximum cut of the corresponding graph is evident. Hence, the decision problem and subsequently the BundleMVL-2 optimization problem are NP-complete.

■

Next, we prove the structural properties satisfied by C^*. To start with, we decompose the revenue function as shown in the following lemma.

Lemma C.2.1. For two sets C and C' such that $C \cap C'$ is the empty set, we have:

$$R_2(C \cup C') = \alpha R_2(C) + (1 - \alpha) T(C, C'), \tag{C.2}$$

where $\alpha = \frac{v_0 + \sum_{i \in C} \sum_{j \in C} \theta_{ij}}{v_0 + \sum_{i \in C \cup C'} \sum_{j \in C \cup C'} \theta_{ij}}$ is a value between 0 and 1, and the function $T(C, C')$ is defined as

$T(C, C') = \frac{\sum_{i \in C'} \sum_{j \in C'} \hat{r}_{ij} \theta_{ij} + 2 \sum_{i \in C} \sum_{j \in C'} \hat{r}_{ij} \theta_{ij}}{\sum_{i \in C'} \sum_{j \in C'} \theta_{ij} + 2 \sum_{i \in C} \sum_{j \in C'} \theta_{ij}}$.

Proof of Lemma C.2.1 .

$$\begin{aligned}
& R_2(C \cup C') \\
&= \frac{\sum_{i \in C \cup C'} \sum_{j \in C \cup C'} \hat{r}_{ij} \theta_{ij}}{v_0 + \sum_{i \in C \cup C'} \sum_{j \in C \cup C'} \theta_{ij}} \\
&= \left(\frac{\sum_{i \in C} \sum_{j \in C} \hat{r}_{ij} \theta_{ij}}{v_0 + \sum_{i \in C} \sum_{j \in C} \theta_{ij}} \right) \left(\frac{v_0 + \sum_{i \in C} \sum_{j \in C} \theta_{ij}}{v_0 + \sum_{i \in C \cup C'} \sum_{j \in C \cup C'} \theta_{ij}} \right) + \\
& \left(\frac{\sum_{i \in C'} \sum_{j \in C'} \hat{r}_{ij} \theta_{ij} + 2 \sum_{i \in C} \sum_{j \in C'} \hat{r}_{ij} \theta_{ij}}{\sum_{i \in C'} \sum_{j \in C'} \theta_{ij} + 2 \sum_{i \in C} \sum_{j \in C'} \theta_{ij}} \right) \left(\frac{\sum_{i \in C'} \sum_{j \in C'} \theta_{ij} + 2 \sum_{i \in C} \sum_{j \in C'} \theta_{ij}}{v_0 + \sum_{i \in C \cup C'} \sum_{j \in C \cup C'} \theta_{ij}} \right) \\
&= \alpha R_2(C) + (1 - \alpha) T(C, C')
\end{aligned}$$

■

Given the above decomposition, the proofs of the Lemmas 4.3.3-4.3.5 are provided below.

Proof of Lemma 4.3.3. Let $i \notin C^*$ such that $r_i > R_2(C^*)$. We know, $R_2(C^* \cup i) = \alpha R_2(C^*) +$

$(1-\alpha)T(C^*,\{i\})$ for some $0 \leq \alpha \leq 1$. As $r_i > R_2(C^*)$, we have $T(C^*,\{i\}) \geq r_i > R_2(C^*)$. As $R_2(C^* \cup i)$ is a convex combination of $R_2(C^*)$ and $T(C^*,\{i\})$, it is greater than $R_2(C^*)$, contradicting the optimality of the recommendation set C^*. ■

Proof of Lemma 4.3.4. Suppose $\exists\ i \in C^*$ such that $r_i + r_j < R_2(C^*)\ \forall\ j \in C^* \backslash i$. This implies that $T(C^* \backslash i, \{i\}) < R_2(C^*)$. We know that $R_2(C^*)$ is a convex combination of $R_2(C^* \backslash i)$ and $T(C^* \backslash i, \{i\})$. Thus, $R_2(C^* \backslash i) > R_2(C^*)$ contradicting the optimality of the recommendation set C^*. ■

Proof of Lemma 4.3.5. Using (C.2), we can decompose the revenue of the revenue-ordered recommendation set A_m as $R_2(A_m) = \alpha R_2(A_{m-1}) + (1-\alpha)T(A_{m-1},\{m\})$, for some $0 \leq \alpha \leq 1$. Also, note that $T(A_{m-1},\{m\}) \geq r_m$. Further, $r_m > R_2(C^*)$ for $m \leq k$. Thus, $T(A_{m-1},\{m\}) > R_2(C^*) \geq R_2(A_{m-1})$ for $m \leq k$. But $R_2(A_m)$ is a convex combination of $T(A_{m-1},\{m\})$ and $R_2(A_{m-1})$. Thus, $R_2(A_m) \geq R_2(A_{m-1})$ for $m \leq k$. ■

C.3. Benchmark Algorithms under the BundleMVL-2 Model

Proof of Lemma 4.4.1. Using the arguments in proof of Lemma 4.3.3 and the local optimality of $\widehat{C}$, we have $\widehat{C}_u \subset \widehat{C}$, where $\widehat{C}_u = \{i : r_i > R_2(\widehat{C})\}$. As $R_2(\widehat{C}) \leq R_2(C^*)$, $C^*_u \subset \widehat{C}_u \subset \widehat{C}$. ■

Algorithm 10 ADXOPTL

Require: Set of all products W, maximum number of removals b, add/delete size parameter l.

1: $C_0 = \phi$, removals$(i) = 0\ \forall i \in W$
2: **repeat**
3: $C^A \leftarrow \arg\max_{C \in \mathcal{C}^A} R(C)$, where $\mathcal{C}^A = \Big\{C : C \in \mathcal{C}, C = C^t \cup C_{add}$ for some $C_{add} \subseteq W$ s.t. $|C_{add}| \leq l,\ removals(i) < b\ \forall i \in C_{add}\Big\}$.
4: $C^D \leftarrow \arg\max_{C \in \mathcal{C}^D} R(C)$, where $\mathcal{C}^D = \Big\{C : C \in \mathcal{C},\ C = C^t \setminus C_{del}$ for some $C_{del} \subset C^t$ s.t. $|C_{del}| \leq l\Big\}$.
5: $C^X \leftarrow \arg\max_{C \in \mathcal{C}^X} R(C)$, where $\mathcal{C}^X = \Big\{C : C \in \mathcal{C},\ C = C_{add} \cup C^t \setminus C_{del}$ for some $C_{del} \subset C^t, C_{add} \subseteq W$ s.t. $|C_{del}| \leq l,\ |C_{add}| \leq l,\ removals(i) < b\ \forall i \in C_{add}\Big\}$.
6: $C^{t+1} = \arg\max \Big\{R(C^A),\ R(C^D),\ R(C^X)\Big\}$.
7: $removals(i) \leftarrow removals(i) + 1\ \forall i \in C^t \backslash C^{t+1}$.
8: **until** $R(C^{t+1}) > R(C^t)$ and $removals(i) < b$ for some $i \in W$.
9: **return** C^t

Proof of Theorem 4.4.3. Let C^* be an optimal recommendation set for the above problem. We will construct a revenue-ordered recommendation set which has revenue $R_2(C^*)$. If C^* is a

revenue-ordered recommendation set, then the theorem is trivially true. Hence, we focus on the case when C^* is not a revenue-ordered recommendation set. Then, there exists products l, m such that $l \in C^*$ and $m \notin C^*$ for some $1 \leq m < l \leq n$.

Let $\tilde{C} = \{m\} \cup C^* \backslash \{l\}$. Thus, we have

$$
\begin{aligned}
&R_2(\tilde{C}) \\
&= \left(\sum_{i \in C^* \backslash \{l\}} r_i V_{\{i\}} + \sum_{i \in C^* \backslash \{l\}} \sum_{j \in C^* \backslash \{l\}, j > i} (r_i + r_j) V_{\{i,j\}} + r_m V_{\{m\}} + \sum_{i \in C^* \backslash \{l\}, i \neq m} (r_i + r_m) V_{\{i,m\}} \right) \\
&\Big/ \left(v_0 + \sum_{i \in \tilde{C}} V_{\{i\}} + \sum_{i \in C^* \backslash \{l\}} \sum_{j \in C^* \backslash \{l\}, j > i} V_{\{i,j\}} + V_{\{m\}} + \sum_{i \in C^* \backslash \{l\}, i \neq m} V_{\{i,m\}} \right) \\
&\geq R_2(C^*).
\end{aligned}
$$

As C^* is an optimal recommendation set, $R_2(\tilde{C}) \leq R_2(C^*)$. Hence, we conclude $R_2(\tilde{C}) = R_2(C^*)$. We can use the above argument repeatedly to construct a sequence of recommendation sets, each having revenue equal to $R_2(C^*)$ until a revenue-ordered recommendation set is obtained. ■

Prior to giving proof of Theorem 4.4.5, we define some notation and lemmas that will be useful. We define $R_{MNL}(C) = \frac{\sum_{i \in C} V_{\{i\}} r_i}{v_0 + \sum_{i \in C} V_{\{i\}}}$, $C^*_{MNL} \in \arg\max_{C \in \mathcal{C}} R_{MNL}(C)$, $C^*_{revord} \in \arg\max_{C \in \{A_1, A_2 \cdots A_n\}} R_2(C)$ and $C^* \in \arg\max_{C \in \mathcal{C}} R(C)$. When $\mathcal{C} = 2^W$, we take C^*_{MNL} to be a revenue-ordered set.

Lemma C.3.1. Under Assumption 4.4.4, $R_{MNL}(C) \leq \frac{2}{2 - \epsilon|C|} R_2(C)$.

Proof of Lemma C.3.1. $R_{MNL}(C) - R_2(C)$

$$
\begin{aligned}
&\leq \frac{\left(\sum_{i,j \in C, j > i} V_{\{i,j\}} \right) \sum_{i \in C} V_{\{i\}} r_i}{\left(v_0 + \sum_{i \in C} V_{\{i\}} \right) \left(v_0 + \sum_{i \in C} V_{\{i\}} + \sum_{i,j \in C, j > i} V_{\{i,j\}} \right)} \\
&\leq \frac{|C|(|C|-1)\, \epsilon \left(\min_{k \in W \cup \phi} V_{\{k\}} \right) \sum_{i \in C} V_{\{i\}} r_i}{2 \left(v_0 + \sum_{i \in C} V_{\{i\}} \right)^2} \\
&\leq \frac{\epsilon |C|}{2} R_{MNL}(C).
\end{aligned}
$$

■

Lemma C.3.2. Under Assumption 4.4.4, $R_2(C) \leq (1 + 2\epsilon|C|)\, R_{MNL}(C)$.

Proof of Lemma C.3.2. $R_2(C) - R_{MNL}(C)$

$$\begin{aligned}
&\leq \frac{\left(v_0 + \sum_{i\in C} V_{\{i\}}\right) \sum_{i,j\in C, j>i} V_{\{i,j\}}(r_i + r_j)}{\left(v_0 + \sum_{i\in C} V_{\{i\}}\right)\left(v_0 + \sum_{i\in C} V_{\{i\}} + \sum_{i,j\in C, j>i} V_{\{i,j\}}\right)} \\
&\leq \frac{2\epsilon|C|\sum_{i\in C} V_{\{i\}} r_i}{v_0 + \sum_{i\in C} V_{\{i\}}} \\
&= 2\epsilon|C|R_{MNL}(C).
\end{aligned}$$

■

Lemma C.3.3. Under Assumption 4.4.4, $R_2(C^*_{MNL}) \geq \frac{2-\epsilon|C^*_{MNL}|}{2+4\epsilon|C^*|} R_2(C^*)$.

Proof of Lemma C.3.3. We have

$$\begin{aligned}
&R_2(C^*) - R_2(C^*_{MNL}) \\
&= (R_2(C^*) - R_{MNL}(C^*)) + (R_{MNL}(C^*) - R_{MNL}(C^*_{MNL})) + (R_{MNL}(C^*_{MNL}) - R_2(C^*_{MNL})) \\
&\leq 2\epsilon|C^*|R_{MNL}(C^*) + \epsilon\frac{|C^*_{MNL}|}{2} R_{MNL}(C^*_{MNL}) \\
&\leq \frac{4|C^*|+|C^*_{MNL}|}{2}\epsilon R_{MNL}(C^*_{MNL}) \\
&\leq \frac{4|C^*|+|C^*_{MNL}|}{2}\frac{2}{2-\epsilon|C^*_{MNL}|}\epsilon R_2(C^*_{MNL}).
\end{aligned}$$

The inequalities are obtained using Lemmas C.3.1 and C.3.2. ■

Proof of Theorem 4.4.5. As C^*_{MNL} is a revenue-ordered recommendation set, using Lemma C.3.3, $R_2(C^*_{revord}) \geq R_2(C^*_{MNL}) \geq \frac{2-\epsilon|C^*_{MNL}|}{2+4\epsilon|C^*|} R_2(C^*) \geq \frac{2-\epsilon n}{2+4\epsilon n} R_2(C^*)$. ■

Proof of Theorem 4.3.2. The unconstrained revenue optimization under the MMC model is given by the following problem:

$$\arg\max_{C\in 2^W} z_1 \sum_{i\in C} r_i \frac{V_{\{i\}}}{V_{\{0\}} + \sum_{k\in C} V_{\{k\}}} + z_2 \sum_{i,j\in C, i<j} (r_i + r_j) \frac{V_{\{i,j\}}}{V_{\{0,0\}} + \sum_{k,l\in C} V_{\{k,l\}}},$$

(2-PRODUCTS MMC AO)

where z_1, z_2 are the probability of purchasing one and two products respectively such that $z_1 + z_2 = 1$. Although Benson et al. (2018) do not model the no purchase option, one way to incorporate the no-purchase option is to assign a utility to the no-purchase option for each size of the subset that can be chosen. The no-purchase option is then treated as an alternative which

is always present irrespective of the recommendation set and is represented with the parameters $V_{\{0\}}$ and $V_{\{0,0\}}$. To establish the NP-completeness of 2-PRODUCTS MMC AO, we use the fact that revenue optimization under the mixture of two multinomial logits (2-CLASS LOGIT AO) is NP-complete (Rusmevichientong et al. 2010b). This optimization problem is:

$$\arg\max_{C\in 2^W} \alpha_1 \sum_{i\in C} s_i \frac{v_i^1}{v_0^1 + \sum_{j\in C} v_j^1} + \alpha_2 \sum_{i\in C} s_i \frac{v_i^2}{v_0^2 + \sum_{j\in C} v_j^2}, \qquad \text{(2-CLASS LOGIT AO)}$$

where the revenues of products are given by $s_1, \cdots, s_n$ with $s_i \in \mathbb{Z}_+ \ \forall\ i \in [n]$, the preference weights are $(v_0^g, v_1^g, \cdots v_n^g)$ with $v_i^g \in \mathbb{Z}_+ \forall\ i \in [n],\ g = 1, 2$, and the probability of a customer belonging to each of the mixture compoenents is (α_1, α_2) with $\alpha_g \in \mathbb{Q}_+,\ g = 1, 2$ and $\alpha_1 + \alpha_2 = 1$. We claim that there is a reduction from a 2-CLASS LOGIT AO instance to a 2-PRODUCTS MMC AO instance and prove this in two steps:

1. Transform an instance of 2-CLASS LOGIT AO to an instance of 2-PRODUCTS MMC AO: Given an instance of 2-CLASS LOGIT AO, we define an instance of 2-PRODUCTS MMC AO by including an additional (n+1)-th product, which we refer to as a *snowball*. The revenues of the products in the transformed instance are same as their original revenue and the snowball has zero revenue i.e. $r_i = s_i\ \forall i \in [n]$ and $r_{n+1} = 0$. The probability of purchasing one and two products is equal to the probability of belonging to each group i.e. $z_g = \alpha_g, g = 1, 2$. The preference weights in the transformed instance are given as:

   (a) $V_i = v_i^1\ \forall\ i \in \{0, 1, \cdots, n\}$ and $V_{n+1} = 0$

   (b) $V_{\{i,j\}} = v_0^2$ if $i = 0, j = 0; v_i^2$ if $j = n+1, i \in [n]; v_j^2$ if $i = n+1, j \in [n]; 0$ else.

2. Given a solution S^* of the above instance of 2-PRODUCTS MMC AO obtain a solution for the original instance of 2-CLASS LOGIT AO: this can be done using Lemma C.3.4.

Thus, any instance of the 2-CLASS LOGIT AO can be reduced to an instance of 2-PRODUCTS MMC AO, proving that 2-PRODUCTS MMC AO is also NP-complete.

■

Lemma C.3.4. If S^* is the solution for the above instance of 2-PRODUCTS MMC AO, then $S^* \backslash \{n+1\}$ is optimal for the 2-CLASS LOGIT AO problem.

Proof of Lemma C.3.4. It is easy to see that $R_{MMC}(S) \leq R_{MMC}(S \cup \{n+1\}) \ \forall \ S \in 2^W$. Thus, without loss of generality, $n+1 \in S^*$. We define $R_{MMNL}(S) = \alpha_1 \sum_{i \in S} s_i \frac{v_i^1}{v_0^1 + \sum_{j \in S} v_j^1} + \alpha_2 \sum_{i \in S} s_i \frac{v_i^2}{v_0^2 + \sum_{j \in S} v_j^2}$. Thus, with the parameters specified as above, $R_{MMNL}(S) = R_{MMC}(S \cup \{n+1\}), \ \forall \ S \in 2^W$. Assume $\hat{S} \neq S^* \backslash \{n+1\}$ is the solution for 2-CLASS LOGIT AO. Thus, $R_{MMC}(\hat{S} \cup \{n+1\}) = R_{MMNL}(\hat{S}) > R_{MMNL}(S^*) = R_{MMC}(S^*)$ contradicting the assumption that S^* is the solution to 2-PRODUCTS MMC AO. ■

C.4. Additional Experiments

Dataset	**Bakery Dataset**			**Kosarak Dataset**		
Model	**#params**	**train_ll**	**test_ll**	**#params**	**train_ll**	**test_ll**
MNL model	50	-91140	-22736	2621	-1317416	-331857
MMC model (0%)	50	-90977	-22691	2621	-1404888	-353659
MMC model (1%)	62	-77674	-19289	2812	-1389932	-350060
MMC model (5%)	110	-77596	-19303	3580	-1386661	-349553
MMC model (20%)	292	-77451	-19373	6460	-1379610	-349239
MMC model (50%)	655	-77214	-19431	12220	-1358675	-350582
MMC model (100%)	1261	-76791	-19571	21819	-1326125	-436080
BundleMVL-2 model	1261	-76791	**-19281**	21733	-1325751	**-294304**
Dataset	**LastFM Genres Dataset**			**YC-items Dataset**		
Model	**#params**	**train_ll**	**test_ll**	**#params**	**train_ll**	**test_ll**
MNL model	443	-1981969	-495887	52677	-256610	-64795
MMC model (0%)	443	-2138089	-534898	52679	-303703	-76847.8
MMC model (1%)	529	-2128021	-532535	52985	-291047	-74024
MMC model (5%)	873	-2112185	-528698	54209	-287511	-73371
MMC model (20%)	2166	-2096532	-525428	58801	-286012	-73212
MMC model (50%)	4750	-2086600	-524088	67985	-285872	-73462
MMC model (100%)	9058	-2076535	-509677	83292	-283825	-73320
BundleMVL-2 model	8871	-2009631	**-493188**	34776	-178693	**-38903**
Dataset	**YC-depts Dataset**			**Instacart Dataset**		
Model	**#params**	**train_ll**	**test_ll**	**#params**	**train_ll**	**test_ll**
MNL model	51	-118387.2	**-29451.3**	5981	-2668742	-670192
MMC model (0%)	53	-137430.7	-34174.4	5981	-2749875	-690323
MMC model (1%)	54	-136112.79	-33852.97	6767	-2722321	-684258
MMC model (5%)	58	-134848.28	-33542.42	9914	-2690143	-679500
MMC model (20%)	75	-134283.9	-33401.2	21714	-2624928	-676805
MMC model (50%)	110	-134251.8	-33392.85	45313	-2510259	-681712
MMC model (100%)	167	-134251.64	-33392.82	84646	-2346723	-1170856
BundleMVL-2 model	134	-137239	-34146	84545	-2346233	**-393099**

Table C.1: Log-likelihood values under different models for six additional datasets.

In Table C.1, we report the fit of the BundleMVL-2 model on six additional datasets, showing that it is extremely competitive with competing models. We benchmark the performance of algorithms against the UCI dataset for: (a) the unconstrained setting in Figure C.1, and (b) the constrained setting in Figure C.2. Results for synthetic datasets are omitted, as they show

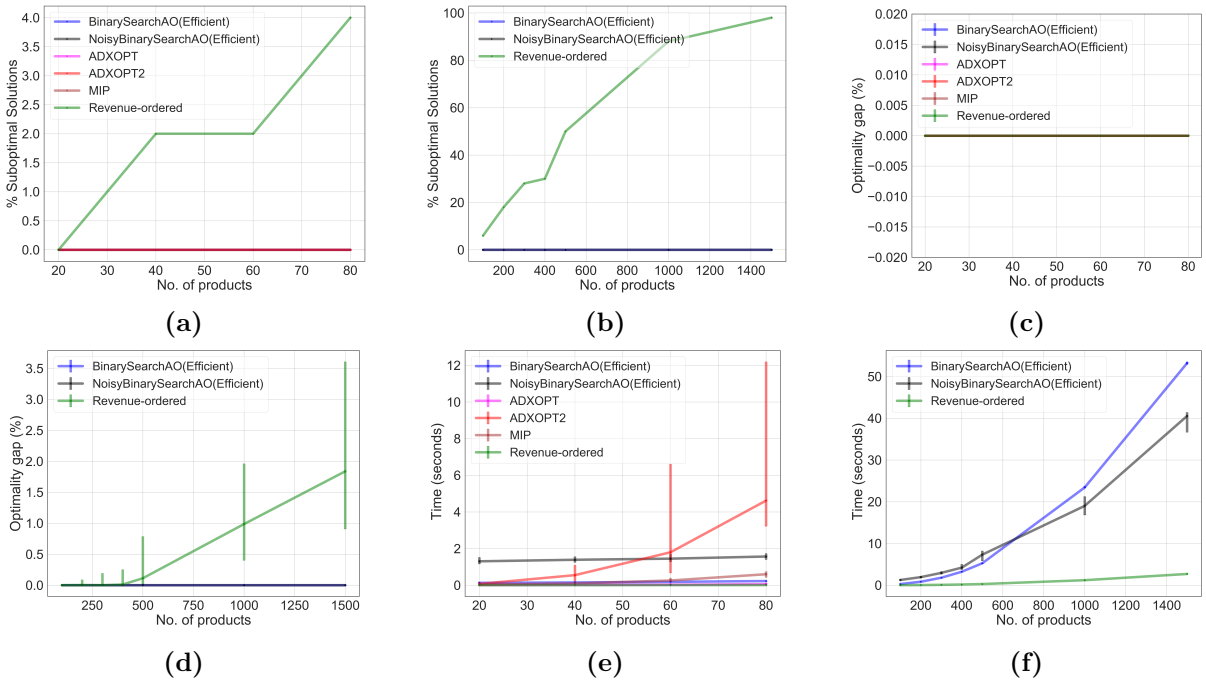


Figure C.1: Optimality and run-time plots on the UCI dataset in the unconstrained setting.

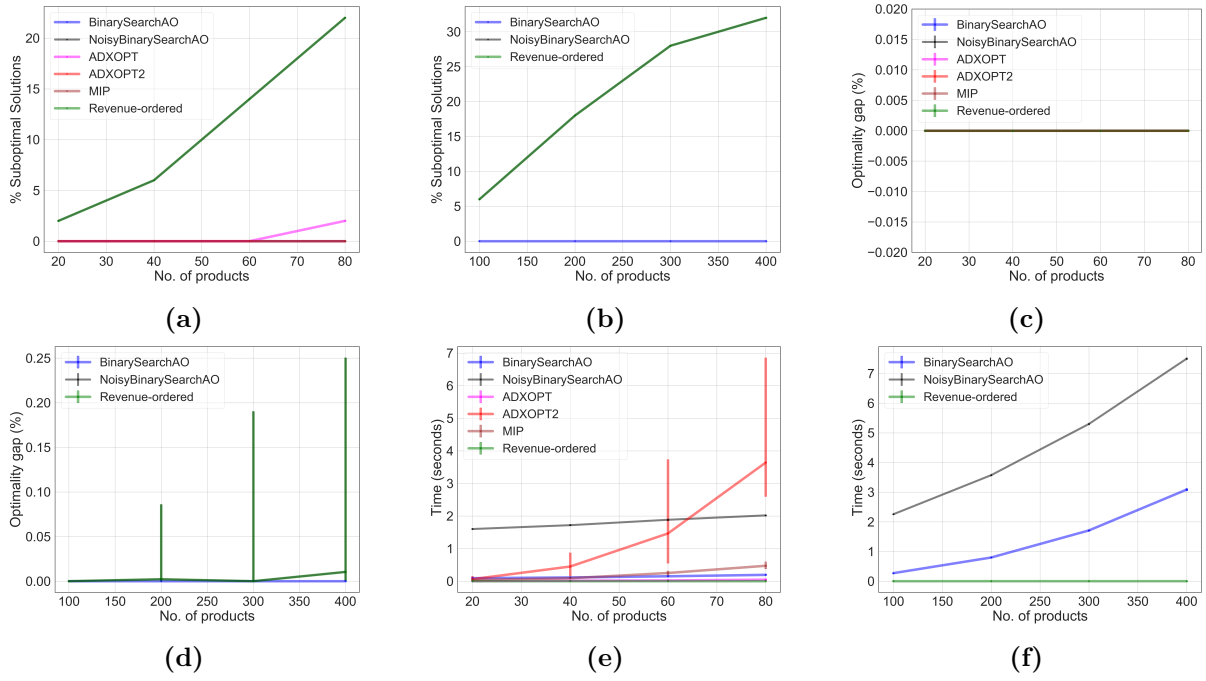
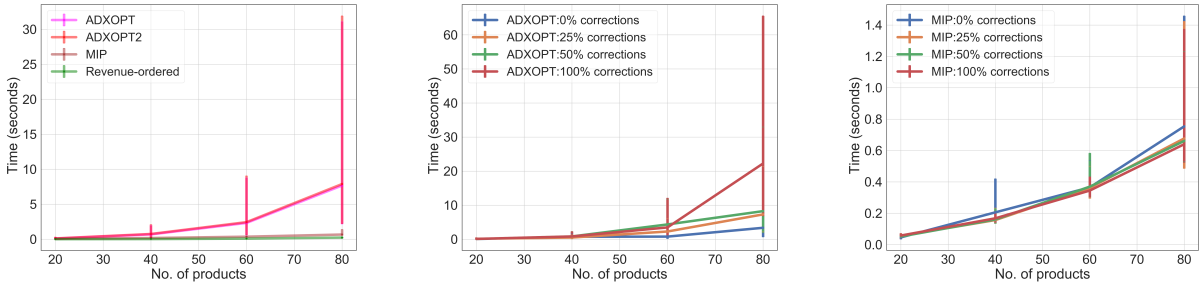


Figure C.2: Optimality and run-time plots on the UCI dataset in the constrained setting.

similar trends.

For our final set of experiments, we focus on optimization using the MMC model, which is not addressed in Benson et al. (2018). We compare a MIP formulation with revenue-ordered, ADXOPT and ADXOPT2 in terms of run-times in Figure C.3a. Next, in Figures C.3b & C.3c, we observe that for a fixed number of products, the time taken by ADXOPT increases as the



(a) **(b)** **(c)**

Figure C.3: For the MMC model: (a): run-time vs products, (b) & (c): run-time vs the number of correction sets.

number of correction sets (H-sets) increase, while it remains unaffected for the MIP.

Appendix D

Appendix to Chapter 5

D.1. Derivation of the Optimization Problem

Here derive the expression for precision used in Section 5.2. Denote the matrix $X \triangleq [x\ Z]$ and $\beta \triangleq [\theta\ \kappa^\top]^\top$. Thus our model is

$$y = X\beta + \epsilon.$$

The least squares estimate $\hat{\beta}$ of β is given by

$$\hat{\beta} = (X^\top X)^{-1}X^\top y = (X^\top X)^{-1}X^\top(X\beta + \epsilon) = \beta + (X^\top X)^{-1}X^\top\epsilon.$$

Then,

$$\mathrm{Var}(\hat{\beta}) = (X^\top X)^{-1}X^\top \mathrm{Var}(\epsilon\epsilon^\top)X(X^\top X)^{-1} = \sigma^2(X^\top X)^{-1}.$$

Thus variance of $\hat{\theta} = \hat{\beta}_1$ is

$$\mathrm{Var}(\hat{\theta}) = \sigma^2 e_1^\top (X^\top X)^{-1} e_1 = \sigma^2 e_1^\top \begin{bmatrix} x^\top x & x^\top Z \\ Z^\top x & Z^\top Z \end{bmatrix}^{-1} e_1 = \frac{\sigma^2}{x^\top(I - Z(Z^\top Z)^{-1}Z^\top)x}.$$

Here, $e_1 \triangleq (1, 0, \ldots)$ is the first coordinate vector, and for the last equality we apply the block matrix inversion formula.

Proof of Proposition 5.2.2. By the Cramér-Rao bound we have that,

$$\mathrm{Cov}\left(\begin{bmatrix} \hat{\theta} \\ \hat{\kappa} \end{bmatrix}\right) \succeq I(\theta, \kappa)^{-1},$$

where $I(\theta, \kappa)$ is the Fisher information matrix. Under the Gaussian assumption for ϵ it is easy to see that,

$$I(\theta, \kappa)^{-1} = \sigma^2 \begin{bmatrix} x^\top x & x^\top Z \\ Z^\top x & Z^\top Z \end{bmatrix}^{-1}.$$

If e_1 is the unit vector along the first coordinate then,

$$\mathrm{Var}(\hat{\theta}) \geq e_1^\top I(\theta, \kappa)^{-1} e_1 = \frac{\sigma^2}{x^\top (I - Z(Z^\top Z)^{-1} Z^\top) x}.$$

Thus,

$$\mathrm{Prec}(\hat{\theta}) \leq \frac{x^\top \left(I - Z(Z^\top Z)^{-1} Z^\top\right) x}{\sigma^2} = \frac{n - x^\top Z(Z^\top Z)^{-1} Z^\top x}{\sigma^2} \leq \frac{n}{\sigma^2}.$$

The inequality follows since $Z(Z^\top Z)^{-1} Z^\top$ is positive semidefinite.

The last statement is consequence of the fact that $x^\top \left(I - Z(Z^\top Z)^{-1} Z^\top\right) x/\sigma^2$ is the precision of the optimal least squares estimator. ■

D.2. Performance of the Randomized Algorithm

We begin with a lemma that relies on some linear algebra arguments.

Lemma D.2.1. Consider a vector $a \in \mathbb{R}^{p-1}$ and an invertible $Q \in \mathbb{R}^{p-1 \times p-1}$ such that the matrix,

$$\begin{bmatrix} 1 & a^\top \\ a & Q \end{bmatrix},$$

is invertible. Then,

$$\begin{bmatrix} 1 & a^\top \end{bmatrix} \begin{bmatrix} 1 & a^\top \\ a & Q \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ a \end{bmatrix} = 1.$$

Proof. By the block matrix inversion formula,

$$\begin{bmatrix} 1 & a^\top \\ a & Q \end{bmatrix}^{-1} = \begin{bmatrix} \rho^{-1} & -\rho^{-1} a^\top Q^{-1} \\ -\rho^{-1} Q^{-1} a & (Q - aa^\top)^{-1} \end{bmatrix} = \begin{bmatrix} \rho^{-1} & -\rho^{-1} a^\top Q^{-1} \\ -\rho^{-1} Q^{-1} a & Q^{-1} + \rho^{-1} Q^{-1} a a^\top Q^{-1} \end{bmatrix},$$

where $\rho \triangleq 1 - a^\top Q^{-1} a$. Thus,

$$\begin{aligned}\begin{bmatrix} 1 & a^\top \end{bmatrix} \begin{bmatrix} 1 & a^\top \\ a & Q \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ a \end{bmatrix} &= \rho^{-1} - 2\rho^{-1} a^\top Q^{-1} a + a^\top Q^{-1} a + \rho^{-1} \left(a^\top Q^{-1} a \right)^2 \\ &= \rho^{-1} - 2\rho^{-1}(1-\rho) + 1 - \rho + \rho^{-1} (1-\rho)^2 \\ &= \frac{1 - 2(1-\rho) + (1-\rho)\rho + 1 + \rho^2 - 2\rho}{\rho} = 1.\end{aligned}$$

■

Now we turn our attention to quantifying the performance of the randomized design.

Lemma D.2.2. Supposed the allocation x is chosen at random from the set $\{\pm 1\}^n$ independently of the covariate values Z, according to some distribution so that, for all $1 \le i < j \le n$,

$$\mathsf{E}_x[x_i x_j] = \alpha,$$

for some constant α. Then,

$$\mathsf{E}_x \left[\mathrm{Loss} \left(\hat{\theta}_x \right) \right] = (1-\alpha)p + \alpha n,$$

where the expectation is taken over the distribution of x.

Proof. Define

$$\overline{Z} \triangleq \frac{1}{n} \sum_{k=1}^{n} Z_k, \qquad \Gamma_n \triangleq Z^\top Z / n,$$

so that (using Lemma D.2.1)

$$Z^\top Z = n\Gamma_n, \qquad \overline{Z}^\top \Gamma_n^{-1} \overline{Z} = 1.$$

Then,

$$\begin{aligned}
\mathsf{E}_x\left[\mathrm{Loss}\left(\hat{\theta}_x\right)\right] &= \mathsf{E}_x\left[x^\top Z(Z^\top Z)^{-1}Zx\right] \\
&= \mathsf{E}_x\left[\left(\sum_{k=1}^n x_k Z_k\right)^\top (n\Gamma_n)^{-1}\left(\sum_{k=1}^n x_k Z_k\right)\right] \\
&= \frac{1}{n}\sum_{k=1}^n\sum_{\ell=1}^n \mathsf{E}_x\left[x_k x_\ell\right] Z_k^\top \Gamma_n^{-1} Z_\ell \\
&= \frac{1}{n}\left(\sum_{k=1}^n Z_k^\top \Gamma_n^{-1} Z_k + \alpha\sum_{k=1}^n\sum_{\ell\neq k} Z_k^\top \Gamma_n^{-1} Z_\ell\right) \\
&= \frac{1-\alpha}{n}\sum_{k=1}^n Z_k^\top \Gamma_n^{-1} Z_k + \frac{\alpha}{n}\left(\sum_{k=1}^n Z_k\right)^\top \Gamma_n^{-1}\left(\sum_{k=1}^n Z_k\right) \\
&= \frac{1-\alpha}{n}\sum_{k=1}^n \mathrm{tr}\left(\Gamma_n^{-1} Z_k Z_k^\top\right) + \alpha n \overline{Z}^\top \Gamma_n^{-1}\overline{Z} \\
&= (1-\alpha)\,\mathrm{tr}\left(\Gamma_n^{-1}\frac{1}{n}\sum_{k=1}^n Z_k Z_k^\top\right) + \alpha n \\
&= (1-\alpha)p + \alpha n.
\end{aligned}$$

■

Proof of Theorem 5.3.2. We can directly apply Lemma D.2.2, with the observation that, under the proposed randomized allocation, if $1 \le i < j \le n$,

$$\alpha \triangleq \mathsf{E}_x\left[x_i x_j\right] = \frac{n/2-1}{n-1} - \frac{n/2}{n-1} = -\frac{1}{n-1}.$$

■

D.3. Asymptotic Performance of the Optimal Design

In this section, we will prove Theorem 5.3.4. The theorem relies on Assumption 5.3.3 with $\Sigma = \rho^2 I$. In particular, we assume that $Z_{i,1} = 1$ and $Z_{i,j} \sim N(0, \rho^2)$ for $j > 1$. Further it is assumed that all entries of Z are independent.

We will place a sequence of problems of dimensions $1 \le p < n$ on the same probability space $(\Omega, \mathcal{F}, \mathbf{P})$. To make the dependence on the dimension clear, we will denote the data matrix by $Z^{n,p}$. In this sequence of data matrices, $Z^{n,p}$ is formed by adding a column to $Z^{n,p}$. The additional column has the distribution $N(0, \rho^2 I_n)$. Let $\{Z^{n,n-1}\}_n$ be an independent sequence. Note that the sequence of matrices $\{Z^{n,p_n}\}_n$ defined using this generative model satisfy the

assumptions laid out in Theorem 5.3.4.

Before we proceed let us set up some notation. Let $\mathrm{Gr}(k, \mathbb{R}^n)$ be the Grassmannian of dimension k in the vector space $\mathbb{R}^n$. In other words, it is the set of all subspaces of dimension k in $\mathbb{R}^n$. Let $\mathcal{S}^{n,p} \in \bigcup_{k=n-p}^{n} \mathrm{Gr}(k, \mathbb{R}^n)$ be the null space of $Z^{n,p\top}$. In other words, it the the orthogonal complement of the span of $Z^{n,p}$. In the following Lemma we show that the $Z^{n,p}$ is full rank.

Lemma D.3.1. The rank of $Z^{n,p}$ is p with probability 1. Thus, $\mathcal{S}^{n,p} \in \mathrm{Gr}(n-p, \mathbb{R}^n)$ almost surely.

Proof. We can prove this inductively. Since $Z^{n,1} = \mathbf{1}$, the statement is trivially true for $p = 1$. Assume that $Z^{n,p-1}$ is rank $p-1$. It implies that the span of $Z^{n,p-1}$ is a $p-1$ dimensional subspace, let us call it $\mathrm{span}(Z^{n,p-1})$. The pth column of $Z^{n,p}$ is non-degenerate Gaussian vector independent of $\mathrm{span}(Z^{n,p-1})$, call it $Z^{,p}$. $\mathsf{P}(Z^{,p} \in \mathrm{span}(Z^{n,p-1})) = 0$. Thus, almost surely, $Z^{n,p}$ is of rank p. ■

From the preceding lemma we can conclude that $\mathcal{S}^{n,n-1}$ is a 1 dimensional subspace, with probability 1. Now we derive an expression for the precision of the optimal estimator for $p = n-1$ in terms of $\mathcal{S}^{n,n-1}$. Let $A \triangleq \{\omega \in \Omega : \mathcal{S}^{n-1}(\omega) \in \mathrm{Gr}(1, \mathbb{R}^n)\}$. From now on, we assume $\Omega = A$ and all subsequent statements hold with probability one.

Consider a function $h : \mathrm{Gr}(1, \mathbb{R}^n) \to \mathbb{R}_+$, such that $h(\mathcal{S}) \triangleq \|y\|_1/\|y\|_2$ for some non-zero $y \in \mathcal{S}$. It is trivial to check that this value is unique for any non-zero y in $\mathcal{S} \in \mathrm{Gr}(1, \mathbb{R}^n)$.

Lemma D.3.2. The precision of the optimal estimator for $p = n-1$ is given by $\sigma^{-2} h(\mathcal{S}^{n,n-1})^2$, almost surely.

Proof. We know that the optimal precision for $n = p-1$ is given by $\sigma^{-2} x^{*\top} P_{Z^{n,n-1\perp}} x^*$, where x^* is the assignment that maximizes (P1). Now note that, $P_{Z^{n,n-1\perp}}$ can be given by $yy^\top/\|y\|_2^2$, for any non-zero $y \in \mathcal{S}^{n,n-1}$. Thus the optimization problem (P1) is,

$$\begin{aligned} \text{maximize} \quad & x^\top \frac{yy^\top}{\|y\|_2^2} x = \frac{(x^\top y)^2}{\|y\|_2^2} \\ \text{subject to} \quad & x \in \{\pm 1\}^n. \end{aligned}$$

But the optimal x is such that $x_i = \mathrm{sgn}(y_i^n)$. With this assignment, the optimal value is $\|y\|_1^2/\|y\|_2^2$. Thus the optimal precision for a given ω is given by $\|y\|_1^2/\sigma^2\|y\|_2^2 = h(\mathcal{S}^{n,n-1})^2/\sigma^2$.

Thus,

$$\text{Prec}_*^{n,n-1} = \frac{h(\mathcal{S}^{n,n-1})^2}{\sigma^2},$$

almost surely. ■

Using the fact that we have all the $Z^{n,p}$s on the same probability space, it is easy to show that the precision monotonically decreases as p grows, for a fixed n.

Lemma D.3.3. For a fixed n, $\text{Prec}_*^{n,p}$ is a decreasing sequence in p. Thus,

$$\inf_{1\le p<n} \frac{\text{Prec}_*^{n,p}}{n} = \frac{\text{Prec}_*^{n,n-1}}{n}$$

Proof. We will prove that $\text{Prec}_*^{n,p}(\omega)$ is a decreasing sequence in p for a fixed n. By construction, $\mathcal{S}^{n,p}(\omega) \subset \mathcal{S}^{n,p-1}(\omega)$. Note that objective value of (P1) can be written as $x^\top P_{\mathcal{S}^{n,p}} x$, where $P_{\mathcal{S}^{n,p}}$ is the projection matrix for the subspace $\mathcal{S}^{n,p}$. For each $x \in \{\pm 1\}^n$ in the constraint set this value will monotonically decrease in p. Thus the optimal value will also decrease with p. This proves that $\text{Prec}_*^{n,p}$ is monotonically decreasing in p. ■

In the light of the preceding lemma we have that,

$$\inf_{1\le p<n} \frac{\text{Prec}_*^{n,p}}{n} = \frac{\text{Prec}_*^{n,n-1}}{n} = \frac{h(\mathcal{S}^{n,n-1})^2}{n\sigma^2}. \tag{D.1}$$

In the last step we find the distribution of $\mathcal{S}^{n,n-1}$. For this purpose let us setup some more notation. Let $\mathcal{Q}^{\mathbf{1}} \subset \mathbb{R}^{n\times n}$ be the group of orthonormal matrices that leave the vector $\mathbf{1} \in \mathbb{R}^n$ invariant. In other words, it is a collection of matrices $Q \in \mathbb{R}^{n\times n}$ that satisfy,

$$QQ^\top = Q^\top Q = I,$$

and

$$Q\mathbf{1} = Q^\top \mathbf{1} = \mathbf{1}.$$

For any $\mathcal{S} \in \text{Gr}(k, \mathbb{R}^n)$, let $Q\mathcal{S} \triangleq \{Qx \mid x \in \mathcal{S}\}$. Let us also define $\mathcal{G}^{\mathbf{1}} \triangleq \{g \in \text{Gr}(1, \mathbb{R}^n) \mid \mathbf{1}^\top P_g \mathbf{1} = 0\}$.

Lemma D.3.4. $Q\mathcal{S}^{n,n-1}$ is distributed as $\mathcal{S}^{n,n-1}$, for any $Q \in \mathcal{Q}^{\mathbf{1}}$. There is a unique distribution on $\mathcal{G}^{\mathbf{1}}$ that has this invariance property. Further it has the same distribution as $\text{span}(\eta^n - \mathbf{1}\bar{\eta}^n)$

with $\eta^n \sim N(0, I_n)$ and $\bar{\eta}^n = n^{-1}\mathbf{1}^\top \eta^n$. Finally $h(\mathcal{S}^{n,n-1})$ has the same distribution as $\|\eta^n - \mathbf{1}\bar{\eta}^n\|_1 / \|\eta^n - \mathbf{1}\bar{\eta}^n\|_2$

Proof. We first show that there is a unique probability distribution on $\mathcal{G}^{\mathbf{1}}$, say μ, such that $\mathcal{S}$ has the same distribution as $Q\mathcal{S}$ for any $Q \in \mathcal{Q}^{\mathbf{1}}$, if $\mathcal{S}$ is distributed as μ. For this purpose we use Theorem 4.1 of James (1954). $\mathcal{Q}^{\mathbf{1}}$ is a transitive compact topological group of transformations of $\mathcal{G}^{\mathbf{1}}$ to itself. Thus by the aforementioned theorem, there exists a unique measure that is invariant under transformations by $Q \in \mathcal{Q}^{\mathbf{1}}$.

Now we prove that $\text{span}(\eta^n - \mathbf{1}\bar{\eta}^n)$ has the specified invariance property. First note that the covariance matrix of $\eta^n - \mathbf{1}\bar{\eta}^n$ is of the form $cI + d\mathbf{1}\mathbf{1}^\top$ for some $c, d \in \mathbb{R}$. Thus the covariance matrix of $Q(\eta^n - \frac{1}{n}\mathbf{1}^\top \eta^n)$ is $Q(cI + d\mathbf{1}\mathbf{1}^\top)Q^\top = cI + d\mathbf{1}\mathbf{1}^\top$. Since both of them are mean 0 and the same covariance matrix, $\text{span}(\eta^n - \mathbf{1}\bar{\eta}^n)$ and $\text{span}(Q(\eta^n - \mathbf{1}\bar{\eta}^n))$ have the same distribution.

By the uniqueness of this distribution μ, we have that $\text{span}(\eta^n - \mathbf{1}\bar{\eta}^n)$ is indeed distributed as $\mathcal{S}^{n,n-1}$. ■

The previous lemma explicitly gives the distribution of $h(\mathcal{S}^{n,n-1})$. Using this, we prove an asymptotic property about $h(\mathcal{S}^{n,n-1})^2/n$.

Lemma D.3.5.

$$\frac{h(\mathcal{S}^{n,n-1})^2}{n} \to \frac{1}{8\pi},$$

in distribution.

Proof. From Lemma D.3.4 we have that, $h(\mathcal{S}^{n,n-1})^2$ has the same distribution as $\frac{\|\eta^n - \mathbf{1}\bar{\eta}^n\|_1^2}{\|\eta^n - \mathbf{1}\bar{\eta}^n\|_2^2}$, with $\eta^n \sim N(0, I_n)$. Further,

$$\begin{aligned}
\frac{\|\eta^n - \mathbf{1}\bar{\eta}^n\|_2^2}{n} &= \frac{1}{n}\sum_{i=1}^n (\eta_i^n - \bar{\eta}^n)^2 \\
&= \frac{1}{n}\sum_{i=1}^n ((\eta^n{}_i)^2 - 2\eta_i^n\bar{\eta}^n + \bar{\eta}^{n2}) \\
&= \frac{1}{n}\sum_{i=1}^n (\eta^n{}_i)^2 - \frac{2}{n}\sum_{i=1}^n \eta_i^n\bar{\eta}^n + (\bar{\eta}^n)^2 \\
&= \frac{1}{n}\sum_{i=1}^n (\eta^n{}_i)^2 - (\bar{\eta}^n)^2
\end{aligned}$$

By strong law of large numbers we have that,

$$\frac{1}{n}\sum_{i=1}^n (\eta^n{}_i)^2 \to 1, \quad \text{almost surely,}$$

and,

$$(\bar{\eta}^n)^2 \to 0, \quad \text{almost surely.}$$

Thus,

$$\frac{1}{\sqrt{n}}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_2 \to 1, \quad \text{a.s.} \tag{D.2}$$

Now we look at $\frac{1}{n}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1$. By triangle inequality,

$$\frac{1}{n}\|\eta^n\| + \frac{1}{n}\|\mathbf{1}\bar{\eta^n}\|_1 \geq \frac{1}{n}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1 \geq \frac{1}{n}\|\eta^n\|_1 - \frac{1}{n}\|\mathbf{1}\bar{\eta^n}\|_1$$

Now by the strong law of large numbers,

$$\frac{1}{n}\|\mathbf{1}\bar{\eta^n}\|_1 = |\bar{\eta^n}| \to 0, \quad \text{a.s.}$$

Thus, $\frac{1}{n}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1$ and $\frac{1}{n}\|\eta^n\|_1$ must have the same limit (if it exists). Again by, strong law of large numbers that,

$$\frac{1}{n}\sum_{i=1}^{n}|\eta_i^n| \to \mathsf{E}[|\xi|] = \frac{1}{2\sqrt{2\pi}},$$

where $\xi \sim N(0,1)$ is a standard normal. Thus,

$$\frac{1}{n}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1 \to \frac{1}{2\sqrt{2\pi}}. \tag{D.3}$$

From (D.2) and (D.3) and using Slutsky's lemma we have that,

$$\frac{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1}{\sqrt{n}\|\eta^n - \mathbf{1}\bar{\eta^n}\|_2} \to \frac{1}{2\sqrt{2\pi}}, \quad \text{almost surely.}$$

By continuity of $x \mapsto x^2$,

$$\frac{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1^2}{n\|\eta^n - \mathbf{1}\bar{\eta^n}\|_2^2} \to \frac{1}{8\pi}, \quad \text{almost surely.} \tag{D.4}$$

Finally by Equation (D.4) and the fact that $h(\mathcal{S}^{n,n-1})^2$ has the same distribution as $\frac{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_1^2}{\|\eta^n - \mathbf{1}\bar{\eta^n}\|_2^2}$,

$$\frac{h(\mathcal{S}^{n,n-1})^2}{n} \to \frac{1}{8\pi},$$

in distribution. ■

Proof of Theorem 5.3.4. Using Lemmas D.3.2 and D.3.3, we have,

$$\frac{\text{Prec}_*^{n,p_n}}{n} \geq \frac{\text{Prec}_*^{n,n-1}}{n} = \frac{h(\mathcal{S}^{n-1})^2}{n\sigma^2}.$$

Finally using Lemma D.3.5 we have,

$$\frac{h(\mathcal{S}^{n-1})^2}{n\sigma^2} \to \frac{1}{8\pi\sigma^2},$$

in distribution. Thus for any $\epsilon > 0$,

$$\mathsf{P}\left(\left|\frac{\text{Prec}_*^{n,n-1}}{n} - \frac{1}{8\pi\sigma^2}\right| > \epsilon\right) \to 0.$$

Therefore,

$$\mathsf{P}\left(\frac{\text{Prec}_*^{n,n-1}}{n} - \frac{1}{8\pi\sigma^2} < -\epsilon\right) \to 0.$$

Finally,

$$\mathsf{P}\left(\frac{\text{Prec}_*^{n,p_n}}{n} - \frac{1}{8\pi\sigma^2} < -\epsilon\right) \to 0.$$

■

D.4. Approximation Guarantee for the Surrogate Problem

We assume without loss that $\Sigma = I$ and begin by establishing a corollary to a basic theorem from the non-asymptotic analysis of random matrices. Let us denote by Γ_n the matrix $\frac{1}{n}Z^\top Z$. Then we have the following approximation result:

Lemma D.4.1. Provided $n \geq \frac{L}{\epsilon^2}\max(p_n, l\log 2/\gamma_n)$, then with probability at least $1-\gamma_n$, we have

$$\|\Gamma_n - I\| \leq \epsilon$$

where L and l are universal constants.

Proof. Let $Z_i^\top$ be a generic row of Z. We first observe that for any x satisfying $\|x\|_2^2 = 1$, we have

$$\mathsf{E}\left(x^\top Z_i\right)^2 = 1$$

so that the rows of Z are isotropic. Moreover, the sub-Gaussian norm of $x^\top Z_i$ is bounded,

uniformly over all x of unit norm, by a universal constant (say, K). This fact follows from a calculation identical to that in equation 5.6 of Vershynin (2012). Consequently, we may apply Theorem 5.39 (specifically see equation 5.23) in Vershynin (2012), so that we have that with probability at least $1 - 2\exp(-c_K s^2)$,

$$\|\Gamma_n - I\| \leq C_K \sqrt{\frac{p_n}{n}} + \frac{s}{\sqrt{n}}$$

where $C_K (\triangleq C)$ and $c_K (\triangleq c)$ depend only on K, and can thus be taken as universal constants. Consequently, if $n \geq \max\left(\frac{4C^2 p_n}{\epsilon^2}, \frac{4\log 2/\gamma_n}{c\epsilon^2}\right)$, then we immediately have the result of the lemma by taking $s = \sqrt{\frac{\log 2/\gamma_n}{c}}$, $L = 4C^2$ and $l = \frac{1}{C^2 c}$. ■

Lemma D.4.1 implies using Lemma 5.36 of Vershynin (2012) (or basic linear algebraic manipulations) that

$$1 - \epsilon \leq \sigma_{\min}\left(\frac{Z}{\sqrt{n}}\right) \leq \sigma_{\max}\left(\frac{Z}{\sqrt{n}}\right) \leq 1 + \epsilon \tag{D.5}$$

with probability at least $1 - \gamma_n$. Here, $\sigma_{\max}$ and $\sigma_{\min}$ are, respectively, minimum and maximum singular values. Now, let us denote by $\hat{\mu}$ the measure over the sequence x_k induced by an optimal solution for the control problem (P3′) and let μ^* denote the measure induced by an optimal policy for our original dynamic optimization problem, (P3). We will demonstrate that an optimal solution to (P3′) is a near optimal solution to (P3). Before doing so, we establish some convenient notation: Denote

$$\bar{\Delta}_n = \begin{bmatrix} \delta_n \\ \Delta_n \end{bmatrix}$$

and recall

$$\Sigma_n \triangleq \frac{1}{n}\sum_{k=1}^{n} Z_{k,2:p_n} Z_{k,2:p_n}^{\top}$$

Proof of Theorem 5.4.2. Now, (D.5) is equivalently stated as:

$$1 - \epsilon \leq \sqrt{\lambda_{\min}(\Gamma_n)} \leq \sqrt{\lambda_{\max}(\Gamma_n)} \leq 1 + \epsilon,$$

with probability at least $1 - \gamma_n$. This, in turn, implies that,

$$\frac{1}{1+\epsilon} \leq \sqrt{\lambda_{\min}\left(\Gamma_n^{-1}\right)} \leq \sqrt{\lambda_{\max}\left(\Gamma_n^{-1}\right)} \leq \frac{1}{1-\epsilon},$$

with probability at least $1 - \gamma_n$. By the Courant-Fisher theorem (see, e.g., Horn and Johnson

2012) we consequently have that,

$$\frac{\|\bar{\Delta}\|_2^2}{(1+\epsilon)^2} \leq \bar{\Delta}^\top \Gamma_n^{-1} \bar{\Delta} \leq \frac{\|\bar{\Delta}\|_2^2}{(1-\epsilon)^2}, \quad \forall\ \bar{\Delta} \in \mathbb{R}^{p_n}, \tag{D.6}$$

with probability at least $1-\gamma_n$.

Now note that

$$\bar{\Delta}^\top \Gamma_n^{-1} \bar{\Delta} = \|\bar{\Delta}\|^2_{\Gamma_n^{-1}} \leq n^2, \tag{D.7}$$

for all feasible values of $\bar{\Delta} \in \mathbb{R}^{p_n}$. This follows from the non-negativity of the objective of (P2), which yields the inequality,

$$n - \frac{\|\bar{\Delta}\|^2_{\Gamma_n^{-1}}}{n} \geq 0.$$

Let A be the set of sample paths such that (D.6) holds. We have that,

$$\begin{aligned}
\mathsf{E}_{\hat{\mu}}\left[\left\|\bar{\Delta}_n\right\|^2_{\Gamma_n^{-1}}\right] &= \mathsf{E}_{\hat{\mu}}\left[\left\|\bar{\Delta}_n\right\|^2_{\Gamma_n^{-1}} \mathbb{I}_A + \left\|\bar{\Delta}_n\right\|^2_{\Gamma_n^{-1}} \mathbb{I}_{A^c}\right] \\
&\leq \frac{\mathsf{E}_{\hat{\mu}}\left[\left\|\bar{\Delta}_n\right\|_2^2\right]}{(1-\epsilon)^2} + n^2 \mathsf{E}_{\hat{\mu}}\left[\mathbb{I}_{A^c}\right] \\
&\leq \frac{\mathsf{E}_{\hat{\mu}}\left[\left\|\bar{\Delta}_n\right\|_2^2\right]}{(1-\epsilon)^2} + \gamma_n n^2 \\
&\leq \frac{\mathsf{E}_{\mu^*}\left[\left\|\bar{\Delta}_n\right\|_2^2\right]}{(1-\epsilon)^2} + \gamma_n n^2
\end{aligned}$$

where the first inequality follows from the right hand side of (D.6) applied to each sample path in A and (D.7) applied to sample paths in A^c. The final inequality follows from the optimality of $\hat{\mu}$ for (P3′). We will now show that

$$\mathsf{E}_{\mu^*}\left[\left\|\bar{\Delta}_n\right\|_2^2\right] \leq (1+\epsilon)^2 \mathsf{E}_{\mu^*}\left[\left\|\bar{\Delta}_n\right\|^2_{\Gamma_n^{-1}}\right] + n^2 p_n \gamma_n + O\left(\sqrt{\frac{n}{p_n - 1}}\right)$$

together with the inequality above, this will yield the theorem. To prove this inequality, we first observe (as we did earlier) that on the set where (D.6) holds, i.e., the set A, $\left\|\bar{\Delta}_n\right\|_2^2 \leq (1+\epsilon)^2 \left\|\bar{\Delta}_n\right\|^2_{\Gamma_n^{-1}}$. Thus,

$$\mathsf{E}_{\mu^*}\left[\left\|\bar{\Delta}_n\right\|_2^2\right] \leq (1+\epsilon)^2 \mathsf{E}_{\mu^*}\left[\left\|\bar{\Delta}_n\right\|^2_{\Gamma_n^{-1}}\right] + \mathsf{E}_{\mu^*}\left[\left\|\bar{\Delta}_n\right\|_2^2 \mathbb{I}_{A^c}.\right]$$

Now note that,

$$\left\|\bar{\Delta}_n\right\|_2^2 = \delta_n^2 + \|\Delta_n\|_2^2 \leq n^2 + \|\Delta_n\|_2^2 .$$

The inequality follows since $|\delta_n| \leq n$. Thus,

$$\begin{aligned}\mathsf{E}_{\mu^*}\left[\left\|\bar{\Delta}_n\right\|_2^2 \mathbb{I}_{A^c}\right] &= n^2\gamma_n + \mathsf{E}_{\mu^*}\left[\|\Delta_n\|_2^2 \mathbb{I}_{A^c}\right] \\ &\leq n^2\gamma_n + \mathsf{E}_{\mu^*}\left[\|\Delta_n\|_2^2 \mathbb{I}_{\left\{\|\Delta_n\|_2^2 \geq \alpha_n(\gamma_n)\right\}}\right].\end{aligned}$$

where $\alpha_n(\gamma_n)$ satisfies $\mathsf{P}_{\mu^*}\left(\|\Delta_n\|^2 \geq \alpha_n(\gamma_n)\right) = \gamma_n$. Applying Lemma D.4.4 yields

$$\mathsf{E}_{\mu^*}\left[\|\Delta_n\|_2^2 \mathbf{1}_{\|\Delta_n\|_2^2 \geq \alpha_n(\gamma_n)}\right] \leq n^2(p_n - 1)\gamma_n + O\left(\sqrt{\frac{n}{p_n - 1}}\right).$$

which yields the result. ■

To complete our proof of the theorem above, we must provide an upper bound on the quantity

$$\mathsf{E}_{\mu^*}\left[\|\Delta_n\|^2 \mathbf{1}_{\|\Delta_n\|^2 \geq \alpha_n(\gamma_n)}\right]$$

where $\alpha_n(\gamma_n)$ satisfies $\mathsf{P}_{\mu^*}\left(\|\Delta_n\|^2 \geq \alpha_n(\gamma_n)\right) = \gamma_n$. In other words $\alpha_n(\gamma_n)$ is the γ_n percentile of $\|\Delta_n\|^2$. Let $\bar{Z}_n$ be a Gamma$(n(p_n - 1)/2, 1)$ random variable, and let $\hat{\alpha}_n(\gamma_n)$ satisfy

$$\mathsf{P}\left(\bar{Z}_n \geq \hat{\alpha}_n(\gamma_n)\right) = \rho.$$

We have

Lemma D.4.2.

$$\mathsf{E}_{\mu^*}\left[\|\Delta_n\|_2^2 \mathbf{1}_{\|\Delta_n\|_2^2 \geq \alpha_n(\gamma_n)}\right] \leq 2n\mathsf{E}\left[\bar{Z}_n \mathbf{1}_{\bar{Z}_n \geq \hat{\alpha}_n(\gamma_n)}\right]$$

Proof. Observe that

$$\begin{aligned}\|\Delta_n\|_2^2 &= \left\|\sum_{k=1}^n x_k Z_{k,2:p_n}\right\|_2^2 \\ &\leq \left(\sum_{k=1}^n \|Z_{k,2:p_n}\|_2\right)^2 \\ &\leq n\sum_{k=1}^n \|Z_{k,2:p_n}\|_2^2 .\end{aligned}$$

where the first inequality follows from the triangle inequality and the second from Cauchy-

Schwartz. We then immediately have that

$$\mathsf{E}_{\mu^*}\left[\|\Delta_n\|_2^2 \mathbf{1}_{\|\Delta_n\|_2^2 \geq \alpha_n(\gamma_n)}\right] \leq n\mathsf{E}\left[\left(\sum_{k=1}^n \|Z_{k,2:p_n}\|_2^2\right)\mathbf{1}_{\sum_{k=1}^n \|Z_{k,2:p_n}\|_2^2 \geq \hat{\alpha}_n(\gamma_n)}\right].$$

But $\frac{1}{2}\sum_{k=1}^n \|Z_{k,2:p_n}\|_2^2 \triangleq \bar{Z}$ is distributed as a Gamma$(n(p_n-1)/2, 1)$ random variable and the claim follows. ■

Now Gamma random variables enjoy the following property on their tails:

Lemma D.4.3. If $\bar{Z} \sim$ Gamma$(k,1)$ and $z(\gamma_k)$ is its ρth quantile (i.e., $z(\gamma_k)$ satisfies $\mathbb{P}(\bar{Z} \geq z(\gamma_k)) = \gamma_k$), then as $k \to \infty$,

$$\mathsf{E}\left[\bar{Z}\mathbf{1}_{\bar{Z}\geq z(\gamma_k)}\right] \leq k\gamma_k + O\left(\frac{1}{\sqrt{k}}\right).$$

Proof. We have:

$$\begin{aligned}
\mathsf{E}\left[\bar{Z}\mathbf{1}_{\bar{Z}\geq z(\gamma_k)}\right] &= \int_{z(\gamma_k)}^\infty z\frac{z^{k-1}\exp(-z)}{\Gamma(k)}dz \\
&= \frac{\Gamma(k+1)}{\Gamma(k)}\int_{z(\gamma_k)}^\infty \frac{z^k\exp(-z)}{\Gamma(k+1)}dz \\
&= k\left[\frac{\Gamma(k+1, z(\gamma_k))}{k\Gamma(k)}\right] \\
&= k\left[\frac{k\Gamma(k, z(\gamma_k)) + z(\gamma_k)^k\exp(-z(\gamma_k)}{k\Gamma(k)}\right] \\
&= k\left[\gamma_k + \frac{z(\gamma_k)^k\exp(-z(\gamma_k))}{k\Gamma(k)}\right]
\end{aligned}$$

where $\Gamma(\cdot,\cdot)$ is the right incomplete Gamma function. The final equality uses the fact that

$$\frac{\Gamma(k, z(\gamma_k))}{\Gamma(k)} = \gamma_k$$

by the definition of $z(\gamma_k)$. But $z^k\exp(-z)/(k\Gamma(k))$ is maximized at $z = k$, so that

$$\frac{z(\gamma_k)^k\exp(-z(\gamma_k))}{k\Gamma(k)} \leq \frac{k^k\exp(-k)}{k\Gamma(k)} = O\left(\frac{1}{k^{3/2}}\right)$$

where we have used Stirling's approximation for $\Gamma(k)$. The result follows. ■

We anticipate that tighter control on the big-oh error term is possible in the above proof, but this level of crudeness suffices. Using the preceding two lemmas now immediately yields:

Lemma D.4.4.

$$\mathsf{E}_{\mu^*}\left[\|\Delta_n\|^2 \mathbf{1}_{\|\Delta_n\|^2 \geq \alpha_n(\gamma_n)}\right] \leq n^2(p_n-1)\rho + O\left(\sqrt{\frac{n}{p_n-1}}\right)$$

D.5. Dynamic Programming Formulation

Proof of Proposition 5.4.3. Consider the following n step Markov decision process (MDP):

1. The state at time k, $S_k = (\delta_{k-1}, \Delta_{k-1}, Z_k)$. The terminal state $S_n = (\delta_n, \Delta_n)$. The state space is $\mathcal{X}_k = \mathbb{R}^{2p}$ for non terminal time periods and is $\mathcal{X}_n = \mathbb{R}^p$ for the terminal time period.

2. The set of actions available to us is $\{\pm 1\}$.

3. At state S_k if action a_k is chosen, the state S_{k+1} is given by $(\delta_{k-1}+a_k, \Delta_{k-1}+a_k Z_{k,2:p}, Z_{k+1})$. After n actions, the terminal state is $S_{n+1} = (\delta_n, \Delta_n)$.

4. There is no per step reward and the terminal reward is $S_{n+1} \mapsto \delta_n^2 + \|\Delta_n\|^2_{\Sigma^{-1}}$.

Note that the MDP is finite horizon and the set of actions available at any point of time is finite, in particular 2. The problem (P3′) is just a terminal cost minimization MDP. It follows from Proposition 4.2.1 in Bertsekas (2013) that a policy x^* that achieves the minimum expected cost. Further there exists a set of functions $J_k^* : \mathcal{X}_k \to \mathbb{R}$ such that $J_k^*(s_k)$ is the cost conditioned on $S_k = s_k$. Trivially,

$$J_{n+1}^*(\delta_n, \Delta_n) = \delta_n^2 + \|\Delta_n\|^2_{\Sigma^{-1}}.$$

These functions follow the recursion,

$$J_k^*(\delta_{k-1}, \Delta_{k-1}, Z_k) = \min_{u \in \{\pm 1\}} \mathsf{E}[J_{k+1}^*(\delta_{k-1}+u, \Delta_{k-1}+uZ_{k,2:p}, Z_{k+1})]. \tag{D.8}$$

Further x_k^*, the optimal policy, has the property that,

$$x_k^* \in \arg\min_{u \in \{\pm 1\}} \mathsf{E}[J_{k+1}^*(\delta_{k-1}+u, \Delta_{k-1}+uZ_{k,2:p}, Z_{k+1})]. \tag{D.9}$$

Let,

$$Q_k(\delta_k, \Delta_k) \triangleq \mathsf{E}\left[J_{k+1}^*(\delta_k, \Delta_k, Z_{k+1})\right]. \tag{D.10}$$

Using (D.8) and (D.10),

$$Q_k(\delta_k, \Delta_k) = \mathsf{E}\left[\min_{u\in\{\pm 1\}} Q_{k+1}(\delta_{k-1} + u, \Delta_{k-1} + uZ_{k,2:p})\right].$$

Further using (D.9) and (D.10),

$$x_k^* \in \arg\min_{u\in\{\pm 1\}} Q_k(\delta_{k-1} + u, \Delta_{k-1} + uZ_{k,2:p}).$$

This proves the dynamic programming proposition. ■

D.6. State Space Collapse

D.6.1. Proof of Theorem 5.4.7

In essence, the proof of Theorem 5.4.7 relies on the symmetry of the elliptical distribution for each covariate vector $Z_{k,2:p}$. In particular, for orthonormal matrix $Q \in \mathbb{R}^{p-1\times p-1}$, $\Sigma^{-1/2}Z_{k,2:p}$ has the same distribution as $Q\Sigma^{-1/2}Z_{k,2:p}$. As a result of this spherical symmetry, under any non-anticipating policy, the distribution of the Mahalanobis distance $\|\Delta_{k+1}\|_{\Sigma^{-1}}$ at time $k+1$ is invariant across all Δ_k of a fixed Mahalanobis distance $\|\Delta_k\|_{\Sigma^{-1}}$ at time k. Thus, as opposed to having to maintain the p-dimensional state variable (δ_k, Δ_k), one merely needs to maintain the two-dimensional state variable $(\delta_k, \|\Delta_k\|_{\Sigma^{-1}})$.

To make this argument formal, we first define an inner product $\langle\cdot,\cdot\rangle_{\Sigma^{-1}}$ on $\mathbb{R}^{p-1}$ by

$$\langle \Delta, \Delta'\rangle_{\Sigma^{-1}} \triangleq \Delta^\top \Sigma^{-1}\Delta',$$

for $\Delta, \Delta' \in \mathbb{R}^{p-1}$. Using the symmetry of elliptical distribution, we can establish that:

Lemma D.6.1. Suppose $\Delta \in \mathbb{R}^{p-1}$ is a fixed $p-1$-dimensional vector and $X \sim \mathrm{Ell}(0, \Sigma, R)$ is an elliptically distributed $p-1$-dimensional random vector. Then,

$$(\langle X, X\rangle_{\Sigma^{-1}}, \langle X, \Delta\rangle_{\Sigma^{-1}}) \overset{d}{=} \left(R^2, R\|\Delta\|_{\Sigma^{-1}}U_1\right).$$

In particular, when $X \sim N(0, \Sigma)$ has a Gaussian distribution, then,

$$(\langle X, X\rangle_{\Sigma^{-1}}, \langle X, \Delta\rangle_{\Sigma^{-1}}) \overset{d}{=} \left(\zeta^\top\zeta, \|\Delta\|_{\Sigma^{-1}}\zeta_1\right),$$

for an independent and normally distributed $p-1$-dimensional random vector $\zeta \sim N(0, I)$.

Proof. Since X follows the elliptical distribution,

$$X \stackrel{d}{=} R\Sigma^{1/2}U.$$

Thus,

$$\langle X, X\rangle_{\Sigma^{-1}} \stackrel{d}{=} R^2 U^\top \Sigma^{1/2}\Sigma^{-1}\Sigma^{1/2}U = R^2.$$

Also,

$$\langle X, \Delta\rangle_{\Sigma^{-1}} \stackrel{d}{=} R\Delta^\top \Sigma^{-1/2}U.$$

But, by the symmetry of the distribution of U, for any $h \in \mathbb{R}^{p-1}$, $h^\top U$ is has the same distribution as $\|h\|_2 U_1$. Due to independence of U and R, $(\langle X, X\rangle_{\Sigma^{-1}}, \langle X, \Delta\rangle_{\Sigma^{-1}})$ is distributed as $(R^2, R\|\Delta\|_{\Sigma^{-1}}U_1)$.

To prove the last statement, note that for the Gaussian case $(R, U) \sim (\|\zeta\|_2, \zeta/\|\zeta\|_2)$, if $\zeta \sim N(0, I)$. Thus,

$$(R^2, R\|\Delta\|_{\Sigma^{-1}}U_1) = \left(\|\zeta\|_2^2, \|\zeta\|_2\|\Delta\|_{\Sigma^{-1}} e_1^\top \frac{\zeta}{\|\zeta\|_2}\right) = (\zeta^\top\zeta, \|\Delta\|_{\Sigma^{-1}}\zeta_1).$$

■

Now we are ready to prove the theorem.

Proof of Theorem 5.4.7. We will prove, by backward induction over $1 \le k \le n$, that

$$Q_k(\delta_k, \Delta_k) = q_k\left(\delta_k, \|\Delta_k\|_{\Sigma^{-1}}^2\right) \tag{D.11}$$

holds for all $\delta_k \in \mathbb{Z}$, $\Delta_k \in \mathbb{R}^{p-1}$. The result will then follow from Proposition 5.4.3.

Comparing (5.9) and (5.10), (D.11) clearly holds for $k = n$.

Now, assume that (D.11) holds for $k+1$. Then, from (5.9),

$$\begin{aligned} Q_k(\delta_k, \Delta_k) &= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q_{k+1}\left(\delta_k + u, \|\Delta_k + uZ_{k+1,2:p}\|^2_{\Sigma^{-1}}\right)\right] \\ &= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q_{k+1}\left(\delta_k + u, \|\Delta_k\|^2_{\Sigma^{-1}} + \|Z_{k+1,2:p}\|^2_{\Sigma^{-1}} + 2u\langle Z_{k+1,2:p}, \Delta_{k+1}\rangle_{\Sigma^{-1}}\right)\right] \\ &= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q_{k+1}\left(\delta_k + u, \|\Delta_k\|^2_{\Sigma^{-1}} + R^2 + 2uRe_1^\top U\|\Delta\|_{\Sigma^{-1}}\right)\right] \\ &\triangleq q_k\left(\delta_k, \|\Delta_k\|^2_{\Sigma^{-1}}\right). \end{aligned} \tag{D.12}$$

The third equality follows from Lemma D.6.1. ■

Finally, we prove Corollary 2.

Proof of Corollary 2. Following the proof of Theorem 5.4.7, we will simplify the expression for (D.12). In particular, using the final part of Lemma D.6.1,

$$\begin{aligned} Q_k(\delta_k, \Delta_k) &= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q^{\text{gauss}}_{k+1}\left(\delta_k + u, \|\Delta_k + uZ_{k+1,2:p}\|^2_{\Sigma^{-1}}\right)\right] \\ &= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q^{\text{gauss}}_{k+1}\left(\delta_k + u, \|\Delta_k\|^2_{\Sigma^{-1}} + R^2 + 2uRe_1^\top U\|\Delta\|_{\Sigma^{-1}}\right)\right] \\ &= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q^{\text{gauss}}_{k+1}\left(\delta_k + u, \|\Delta_k\|^2_{\Sigma^{-1}} + \zeta^\top\zeta + 2u\zeta_1\|\Delta\|_{\Sigma^{-1}}\right)\right] \\ &= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q^{\text{gauss}}_{k+1}\left(\delta_k + u, \|\Delta_k\|^2_{\Sigma^{-1}} + \xi + \eta^2 + 2u\eta\|\Delta\|_{\Sigma^{-1}}\right)\right] \\ &= \mathsf{E}\left[\min_{u\in\{\pm 1\}} \; q^{\text{gauss}}_{k+1}\left(\delta_k + u, (\|\Delta_k\|_{\Sigma^{-1}} + u\eta)^2 + \xi\right)\right]. \end{aligned}$$

Here, $\xi \sim \chi^2_{p-2}$ if $p > 2$ and $\xi \triangleq 0$ if $p = 2$, and $\eta \sim N(0,1)$ are independent of each other. ■

Bibliography

Abramowitz, Milton, Irene A Stegun. 1948. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, vol. 55. US Government printing office.

Adomavicius, Gediminas, Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* **17**(6) 734–749.

Agrawal, S., V. Avadhanula, V. Goyal, A. Zeevi. 2016. A near-optimal exploration-exploitation approach for assortment selection. *Proceedings of the 2016 ACM Conference on Economics and Computation (EC)* 599–600.

Agrawal, S., N. Goyal. 2017a. Near-optimal regret bounds for thompson sampling. *J. ACM* **64**(5).

Agrawal, Shipra, Vashist Avadhanula, Vineet Goyal, Assaf Zeevi. 2017. Thompson sampling for the mnl-bandit. *Conference on Learning Theory*. 76–78.

Agrawal, Shipra, Nikhil R. Devanur. 2014. Bandits with concave rewards and convex knapsacks. *Proceedings of the Fifteenth ACM Conference on Economics and Computation*. EC '14, Association for Computing Machinery, New York, NY, USA, 989–1006.

Agrawal, Shipra, Navin Goyal. 2017b. Near-optimal regret bounds for thompson sampling. *Journal of the ACM (JACM)* **64**(5) 1–24.

Akamai. 2017. Akamai online retail performance report: Milliseconds are critical. https://www.akamai.com/uk/en/about/news/press/2017-press/akamai-releases-spring-2017-state-of-online-retail-performance-report.jsp.

Amani, Sanae, Mahnoosh Alizadeh, Christos Thrampoulidis. 2020. Generalized linear bandits with safety constraints. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3562–3566.

Amazon. 2019. Amazon auto-targeting. URL https://tinyurl.com/yx9lyfwq.

Anandkumar, Animashree, Nithin Michael, Ao Kevin Tang, Ananthram Swami. 2011. Distributed algorithms for learning and cognitive medium access with logarithmic regret. *IEEE Journal on Selected Areas in Communications* **29**(4) 731–745.

Andoni, Alexandr, Piotr Indyk. 2008. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM* **51**(1) 117.

Andoni, Alexandr, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, Ludwig Schmidt. 2015. Practical and optimal lsh for angular distance. *Advances in Neural Information Processing Systems*. 1225–1233.

Andoni, Alexandr, Ilya Razenshteyn. 2015. Optimal data-dependent hashing for approximate near neighbors. *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*. ACM, 793–801.

Atkinson, A. C. 1982. Optimum biased coin designs for sequential clinical trials with prognostic factors. *Biometrika* **69**(1) 61–67.

Atkinson, A. C. 1999. Optimum biased-coin designs for sequential treatment allocation with covariate information. *Statistics in Medicine* **18**(14) 1741–1752.

Atkinson, A. C. 2002. The comparison of designs for sequential clinical trials with covariate information. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **165**(2) 349–373.

Atkinson, A. C. 2014. Selecting a biased-coin design. *Statistical Science* **29**(1) 144–163.

Auer, Peter, Nicolo Cesa-Bianchi, Yoav Freund, Robert E Schapire. 2002a. The nonstochastic multiarmed bandit problem. *SIAM journal on computing* **32**(1) 48–77.

Auer, Peter, Nicolò Cesa-Bianchi, Yoav Freund, Robert E. Schapire. 2002b. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* **32**(1) 48–77.

Badanidiyuru, Ashwinkumar, Robert Kleinberg, Aleksandrs Slivkins. 2018. Bandits with knapsacks. *J. ACM* **65**(3) 13:1–13:55. doi:10.1145/3164539. URL http://doi.acm.org/10.1145/3164539.

Baldi Antognini, A., M. Zagoraiou. 2011. The covariate-adaptive biased coin design for balancing clinical trials in the presence of prognostic factors. *Biometrika* **98**(3) 519–535.

Ball, F. G., A. F. M. Smith, I. Verdinelli. 1993. Biased coin designs with a Bayesian bias. *Journal of Statistical Planning and Inference* **34**(3) 403–421.

Ben-Tal, A., A. Nemirovski. 2001. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Society for Industrial and Applied Mathematics.

Benson, Austin R, Ravi Kumar, Andrew Tomkins. 2018. A discrete choice model for subset selection. *Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 37–45.

Bentley, Jon Louis. 1975. Multidimensional binary search trees used for associative searching. *Communications of the ACM* **18**(9) 509–517.

Bertsekas, D. P. 2013. *Abstract Dynamic Programming*. Athena Scientific, Belmont, MA.

Bertsimas, D., M. Johnson, N. Kallus. 2015. The power of optimization over randomization in designing experiments involving small samples. *Operations Research* **63**(4) 868–876.

Besag, Julian. 1974. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)* **36**(2) 192–225.

Blackwell, D., J. L. Hodges. 1957. Design for the control of selection bias. *The Annals of Mathematical Statistics* **28**(2) 449–460.

Blanchet, Jose, Guillermo Gallego, Vineet Goyal. 2016. A Markov chain approximation to choice modeling. *Operations Research* **64**(4) 886–905.

Bonmin. 2019. Bonmin solver. https://www.coin-or.org/Bonmin/. Accessed: 2020-05-16.

Boros, Endre, Peter L Hammer, Gabriel Tavares. 2007. Local search heuristics for quadratic unconstrained binary optimization (QUBO). *Journal of Heuristics* **13**(2) 99–132.

Bubeck, Sébastien, Nicolò Cesa-Bianchi. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* **5**(1) 1–122.

Burnashev, Marat Valievich, Kamil'Shamil'evich Zigangirov. 1974. An interval estimation problem for controlled observations. *Problemy Peredachi Informatsii* **10**(3) 51–61.

Cambanis, S., S. Huang, G. Simons. 1981. On the theory of elliptically contoured distributions. *Journal of Multivariate Analysis* **11**(3) 368–385.

Canlas, Marnelli, Kristoffer Paolo Cruz, Ma Kristle Dimarucut, Patrick Uyengco, Gregory Tangonan, Ma Leonora Guico, Nathaniel Libatique, Cesar Pineda. 2010. A quantitative analysis of the quality of service of short message service in the philippines. *2010 IEEE International Conference on Communication Systems*. IEEE, 710–714.

Cao, Wei, Jian Li, Yufei Tao, Zhize Li. 2015. On top-k selection in multi-armed bandits and hidden bipartite graphs. *Advances in Neural Information Processing Systems*. 1036–1044.

Cesa-Bianchi, Nicolo, Claudio Gentile, Yishay Mansour. 2014. Regret minimization for reserve prices in second-price auctions. *IEEE Transactions on Information Theory* **61**(1) 549–564.

Charikar, Moses S. 2002. Similarity estimation techniques from rounding algorithms. *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*. ACM, 380–388.

Chávez, Edgar, Gonzalo Navarro, Ricardo Baeza-Yates, José Luis Marroquín. 2001. Searching in metric spaces. *ACM computing surveys (CSUR)* **33**(3) 273–321.

Chen, Lijie, Anupam Gupta, Jian Li. 2016. Pure exploration of multi-armed bandit under matroid constraints. *Conference on Learning Theory*. 647–669.

Chen, Shouyuan, Tian Lin, Irwin King, Michael R Lyu, Wei Chen. 2014. Combinatorial pure exploration of multi-armed bandits. *Advances in Neural Information Processing Systems*. 379–387.

Chick, S., M. Forster, P. Pertile. 2017. A Bayesian decision theoretic model of sequential experimentation with delayed response. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **79**(5) 1439–1462.

Chick, S. E., P. Frazier. 2012. Sequential sampling with economics of selection procedures. *Management Science* **58**(3) 550–569.

Chick, S. E., N. Gans. 2009. Economic analysis of simulation selection problems. *Management Science* **55**(3) 421–437.

Clement, J. 2019. Worldwide retail e-commerce sales. https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/. Accessed: 2020-05-16.

Cook, T. D., D. T. Campbell, A. Day. 1979. *Quasi-Experimentation: Design & Analysis Issues for Field Settings*. Houghton Mifflin Boston.

Cox, David R. 1972. The analysis of multivariate binary data. *Applied Statistics* 113–120.

Das, Abhinandan S, Mayur Datar, Ashutosh Garg, Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. *Proceedings of the 16th international conference on World Wide Web*. 271–280.

Daulton, Samuel, Shaun Singh, Vashist Avadhanula, Drew Dimmery, Eytan Bakshy. 2019. Thompson sampling for contextual bandit problems with auxiliary safety constraints. *arXiv preprint arXiv:1911.00638* .

Davis, James, Guillermo Gallego, Huseyin Topaloglu. 2013. Assortment planning under the multinomial logit model with totally unimodular constraint structures. *Department of IEOR, Columbia University. Available at http://www. columbia. edu/ gmg2/logit_const. pdf* .

Désir, Antoine, Vineet Goyal, Danny Segev. 2016. Assortment optimization under a random swap based distribution over permutations model. *EC*. 341–342.

Désir, Antoine, Vineet Goyal, Danny Segev, Chun Ye. 2015. Capacity constrained assortment optimization under the markov chain based choice model. *Operations Research, Forthcoming* .

Drugan, Madalina M, Ann Nowe. 2013. Designing multi-objective multi-armed bandits algorithms: A study. *The 2013 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

Dunning, Iain, Swati Gupta, John Silberholz. 2018. What works best when? a systematic evaluation of heuristics for max-cut and qubo. *INFORMS Journal on Computing* **30**(3) 608–624.

Efron, B. 1971. Forcing a sequential experiment to be balanced. *Biometrika* **58**(3) 403–417.

Ettl, Markus, Pavithra Harsha, Anna Papush, Georgia Perakis. 2020. A data-driven approach to personalized bundle pricing and recommendation. *Manufacturing & Service Operations Management* **22**(3) 461–480.

Facebook. 2016. Facebook targeting expansion. URL https://tinyurl.com/y3ss2j8g.

Farias, Vivek F, Srikanth Jagabathula, Devavrat Shah. 2013. A nonparametric approach to modeling choice with limited data. *Management Science* **59**(2) 305–322.

Feldman, Jacob, Dennis Zhang, Xiaofei Liu, Nannan Zhang. 2019. Customer choice models versus machine learning: Finding optimal product displays on alibaba. *SSRN 3232059* .

Fisher, R. A. 1935. *The Design of Experiments*. Oliver & Boyd.

Freund, Yoav, Raj Iyer, Robert E Schapire, Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *Journal of machine learning research* **4**(Nov) 933–969.

Galichet, Nicolas, Michele Sebag, Olivier Teytaud. 2013. Exploration vs exploitation vs safety: Risk-aware multi-armed bandits. *Asian Conference on Machine Learning*. 245–260.

Glover, Fred, Gary Kochenberger. 2018. A tutorial on formulating QUBO models. *ArXiv 1811.11538* .

Goemans, M. X., D. P. Williamson. 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM* **42**(6) 1115–1145.

Google. 2014. Google auto-targeting. URL https://tinyurl.com/y3c4bdaj.

Grbovic, Mihajlo, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, Doug Sharp. 2015. E-commerce in your inbox: Product recommendations at scale. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1809–1818.

Han, B., N. H. Enas, D. McEntegart. 2009. Randomization by minimization for unbalanced treatment allocation. *Statistics in Medicine* **28**(27) 3329–3346.

Han, Shaoning, Andrés Gómez, Oleg A Prokopyev. 2019. Assortment optimization and submodularity .

Hauser, J. R., G. L. Urban, G. Liberali, M. Braun. 2009. Website morphing. *Marketing Science* **28**(2) 202–223.

Horn, R. A., C. R. Johnson. 2012. *Matrix Analysis*. Cambridge University Press.

Hruschka, Harald, Martin Lukanowicz, Christian Buchta. 1999. Cross-category sales promotion effects. *Journal of Retailing and Consumer Services* **6**(2) 99–105.

Hu, Y., F. Hu. 2012. Asymptotic properties of covariate-adaptive randomization. *The Annals of Statistics* **40**(3) 1794–1815.

Huber, Mark. 2013. Nearly optimal bernoulli factories for linear functions. *arXiv preprint arXiv:1308.1562* .

Immorlica, N., K. A. Sankararaman, R. Schapire, A. Slivkins. 2019. Adversarial bandits with knapsacks. *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. 202–219.

Immorlica, Nicole, Brendan Lucier, Jieming Mao, Vasilis Syrgkanis, Christos Tzamos. 2018. Combinatorial assortment optimization. *International Conference on Web and Internet Economics*. Springer, 218–231.

Indyk, Piotr, Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 604–613.

Jagabathula, Srikanth. 2014. Assortment optimization under general choice. *SSRN 2512831* .

James, A. T. 1954. Normal multivariate analysis and the orthogonal group. *The Annals of Mathematical Statistics* 40–75.

Jamieson, Kevin, Robert Nowak. 2014. Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting. *2014 48th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 1–6.

Jin, Rong, Luo Si, ChengXiang Zhai. 2002. Preference-based graphic models for collaborative filtering. *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 329–336.

Jin, Rong, Luo Si, ChengXiang Zhai, Jamie Callan. 2003. Collaborative filtering with decoupled models for preferences and ratings. *Proceedings of the twelfth international conference on Information and knowledge management*. ACM, 309–316.

Johari, R., L. Pekelis, D. J. Walsh. 2017. Always valid inference: Bringing sequential analysis to A/B testing. Working paper.

Juan, Yuchin, Yong Zhuang, Wei-Sheng Chin, Chih-Jen Lin. 2016. Field-aware factorization machines for ctr prediction. *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 43–50.

Kallus, N. 2013. Regression-robust designs of controlled experiments. Working paper.

Kapelner, A., A. Krieger. 2014. Matching on-the-fly: Sequential allocation with higher power and efficiency. *Biometrics* **70**(2) 378–388.

Karatzoglou, Alexandros, Alex Smola, Markus Weimer. 2010. Collaborative filtering on a budget. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 389–396.

Kasy, M. 2013. Why experimenters should not randomize, and what they should do instead. *European Economic Association & Econometric Society* .

Katz-Samuels, Julian, Clayton Scott. 2019. Top feasible arm identification. *The 22nd International Conference on Artificial Intelligence and Statistics*. 1593–1601.

Kazerouni, Abbas, Mohammad Ghavamzadeh, Yasin Abbasi Yadkori, Benjamin Van Roy. 2017. Conservative contextual linear bandits. *Advances in Neural Information Processing Systems*. 3910–3919.

Keane, MS, George L O'Brien. 1994. A bernoulli factory. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **4**(2) 213–219.

Kim, S.-H., B. L. Nelson. 2006. Selecting the best system. *Handbooks in operations research and management science* **13** 501–534.

Kleinberg, Jon, Sendhil Mullainathan, Johan Ugander. 2017. Comparison-based choices. *Proceedings of the 2017 ACM Conference on Economics and Computation*. 127–144.

Kök, A Gürhan, Marshall L Fisher, Ramnath Vaidyanathan. 2008. Assortment planning: Review of literature and industry practice. *Retail supply chain management*. Springer, 99–153.

Koningstein, Ross. 2006. Suggesting and/or providing targeting information for advertisements. US Patent App. 11/026,508.

Kopalle, Praveen K, Aradhna Krishna, Joao L Assuncao. 1999. The role of market expansion on equilibrium bundling strategies. *Managerial and Decision Economics* **20**(7) 365–377.

Kunaver, Matevž, Tomaž Požrl. 2017. Diversity in recommender systems–a survey. *Knowledge-Based Systems* **123** 154–162.

Kuznetsova, O. M., Y. Tymofyeyev. 2012. Preserving the allocation ratio at every allocation with biased coin randomization and minimization in studies with unequal allocation. *Statistics in Medicine* **31**(8) 701–723.

Langford, J., T. Zhang. 2007. The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in Neural Information Processing Systems*. 1096–1103.

Langford, John, Tong Zhang. 2008. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems*. 817–824.

Li, Lihong, Shunbao Chen, Jim Kleban, Ankur Gupta. 2015. Counterfactual estimation and optimization of click metrics in search engines: A case study. *Proceedings of the 24th International Conference on World Wide Web*. 929–934.

Li, Lihong, Wei Chu, John Langford, Xuanhui Wang. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *Proceedings of the fourth ACM international conference on Web search and data mining*. 297–306.

Liu, Han, Xiangnan He, Fuli Feng, Liqiang Nie, Rui Liu, Hanwang Zhang. 2018. Discrete factorization machines for fast feature-based recommendation. *arXiv preprint arXiv:1805.02232* .

Liu, Xianglong, Junfeng He, Cheng Deng, Bo Lang. 2014. Collaborative hashing. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2139–2146.

Manchanda, Puneet, Asim Ansari, Sunil Gupta. 1999. The shopping basket: A model for multicategory purchase incidence decisions. *Marketing Science* **18**(2) 95–114.

MarketWatch. 2020. Marketwatch a2p report. URL `https://rb.gy/0w96oi`.

May, B. C., N. Korda, A. Lee, D. S. Leslie. 2012. Optimistic bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research* **(13)** 2069–2106.

McCardle, Kevin F, Kumar Rajaram, Christopher S Tang. 2007. Bundling retail products: Models and analysis. *European Journal of Operational Research* **177**(2) 1197–1217.

Meng, Xiaoqiao, Petros Zerfos, Vidyut Samanta, Starsky HY Wong, Songwu Lu. 2007. Analysis of the reliability of a nationwide short message service. *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. IEEE, 1811–1819.

Mikolov, Tomas, Kai Chen, Greg Corrado, Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*. 3111–3119.

Mustafa, Nabil H, Rajiv Raman, Saurabh Ray. 2014. Settling the apx-hardness status for geometric set cover. *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*. IEEE, 541–550.

Nemhauser, George L, Laurence A Wolsey, Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming* **14**(1) 265–294.

Nesterov, Y. 1997. Semidefinite relaxation and nonconvex quadratic optimization. Tech. rep., Université Catholique de Louvain, Center for Operations Research and Econometrics.

Oliver, C., L. Li. 2011. An empirical evaluation of thompson sampling. *In Advances in Neural Information Processing Systems (NIPS)* **24** 2249?2257.

Omohundro, Stephen M. 1989. *Five balltree construction algorithms*. International Computer Science Institute Berkeley.

Outbrain. 2017. Similar tech. URL https://www.similartech.com/technologies/outbrain. [Online; accessed April 13, 2017].

Ovum. 2017. Sustaining a2p sms growth while securing mobile network. URL https://rb.gy/qqonzd.

Palmer, Oliver. 2016. How does page load time impact engagement? https://blog.optimizely.com/2016/07/13/how-does-page-load-time-impact-engagement/. Accessed: 2020-06-14.

Pardalos, Panos M, Somesh Jha. 1992. Complexity of uniqueness and local search in quadratic 0–1 programming. *Operations Research Letters* **11**(2) 119–123.

Paria, Biswajit, Kirthevasan Kandasamy, Barnabás Póczos. 2018. A flexible framework for multi-objective bayesian optimization using random scalarizations. *arXiv preprint arXiv:1805.12168* .

Paulevé, Loïc, Hervé Jégou, Laurent Amsaleg. 2010. Locality sensitive hashing: A comparison of hash function types and querying mechanisms. *Pattern Recognition Letters* **31**(11) 1348–1358.

Plackett, Robin L. 1975. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **24**(2) 193–202.

Pocock, S. J., R. Simon. 1975. Sequential treatment assignment with balancing for prognostic factors in the controlled clinical trial. *Biometrics* 103–115.

Pukelsheim, F. 2006. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics.

Rader Jr, David J, Gerhard J Woeginger. 2002. The quadratic 0–1 knapsack problem with series–parallel support. *Operations Research Letters* **30**(3) 159–166.

Raudenbush, S. W., A. Martinez, J. Spybrook. 2007. Strategies for improving precision in group-randomized experiments. *Educational Evaluation and Policy Analysis* **29**(1) 5–29.

Rendle, Steffen. 2010. Factorization machines. *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 995–1000.

Robbins, Herbert. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society* **58**(5) 527–535.

Rosenberger, W. F., O. Sverdlov. 2008. Handling covariates in the design of clinical trials. *Statistical Science* 404–419.

Rusmevichientong, Paat, Zuo-Jun Max Shen, David B Shmoys. 2010a. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations research* **58**(6) 1666–1680.

Rusmevichientong, Paat, David Shmoys, Chaoxu Tong, Huseyin Topaloglu. 2014. Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management* **23**(11) 2023–2039.

Rusmevichientong, Paat, David Shmoys, Huseyin Topaloglu. 2010b. Assortment optimization with mixtures of logits. Tech. rep., School of IEOR, Cornell University.

Russell, Gary J, Ann Petersen. 2000. Analysis of cross category dependence in market basket selection. *Journal of Retailing* **76**(3) 367–392.

Sankararaman, Abishek, Ayalvadi Ganesh, Sanjay Shakkottai. 2019. Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* **3**(3) 1–35.

Schapire, William W Cohen Robert E, Yoram Singer. 1998. Learning to order things. *Advances in Neural Information Processing Systems* **10**(451) 24.

Schwartz, E. M., E. T. Bradlow, P. S. Fader. 2017. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science* .

Scott, Steven L. 2010. A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry* **26**(6) 639–658.

Seetharaman, PB, Siddhartha Chib, Andrew Ainslie, Peter Boatwright, Tat Chan, Sachin Gupta, Nitin Mehta, Vithala Rao, Andrei Strijnev. 2005. Models of multi-category choice behavior. *Marketing Letters* **16**(3-4) 239–254.

Singh, Vishal P, Karsten T Hansen, Sachin Gupta. 2005. Modeling preferences for common attributes in multicategory brand choice. *Journal of Marketing Research* **42**(2) 195–209.

Sinha, Deeksha, Theja Tulabandhula. 2020. Optimizing revenue while showing relevant assortments at scale. *ArXiv 2003.04736* .

Slivkins, Aleksandrs. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning* **12**(1-2) 1–286.

Slivkins, Aleksandrs, Jennifer Wortman Vaughan. 2014. Online decision making in crowdsourcing markets: Theoretical challenges. *ACM SIGecom Exchanges* **12**(2) 4–23.

Smith, R. L. 1984a. Properties of biased coin designs in sequential clinical trials. *The Annals of Statistics* 1018–1034.

Smith, R. L. 1984b. Sequential treatment allocation using biased coin designs. *Journal of the Royal Statistical Society. Series B (Methodological)* 519–543.

Sproull, Robert F. 1991. Refinements to nearest-neighbor searching in k-dimensional trees. *Algorithmica* **6**(1) 579–589.

Steensma, D. P., H. M. Kantarjian. 2014. Impact of cancer research bureaucracy on innovation, costs, and patient care. *Journal of Clinical Oncology* **32**(5) 376–378.

Talluri, Kalyan, Garrett Van Ryzin. 2004. Revenue management under a general discrete choice model of consumer behavior. *Management Science* **50**(1) 15–33.

Terasawa, Kengo, Yuzuru Tanaka. 2007. Spherical lsh for approximate nearest neighbor search on unit hypersphere. *Workshop on Algorithms and Data Structures*. Springer, 27–38.

Thompson, W.R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* **25**(3/4) 285–294.

Toubia, O., J. R. Hauser, D. I. Simester. 2004. Polyhedral methods for adaptive choice-based conjoint analysis. *Journal of Marketing Research* **41**(1) 116–131.

Toubia, O., D. I. Simester, J. R. Hauser, E. Dahan. 2003. Fast polyhedral adaptive conjoint estimation. *Marketing Science* **22**(3) 273–303.

Train, Kenneth E. 2009. *Discrete choice methods with simulation*. Cambridge University Press.

Tran-Thanh, Long, Archie Chapman, Alex Rogers, Nicholas R Jennings. 2012. Knapsack based optimal policies for budget–limited multi–armed bandits. *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Twilio, Uber. 2020. Uber built a great ridesharing experience with sms & voice. URL https://customers.twilio.com/208/uber/.

Vershynin, R. 2012. Introduction to the non-asymptotic analysis of random matrices. Y. Eldar, G. Kutyniok, eds., *Compressed Sensing, Theory and Applications*. Cambridge University Press, 210–268.

Williams, Huw CWL. 1977. On the formation of travel demand models and economic evaluation measures of user benefit. *Environment and planning A* **9**(3) 285–344.

Woodroofe, M. 1979. A one-armed bandit problem with a concomitant variable. *Journal of the American Statistical Association* **74**(368) 799–806.

Wu, Yifan, Roshan Shariff, Tor Lattimore, Csaba Szepesvári. 2016. Conservative bandits. *International Conference on Machine Learning*. 1254–1262.

Yahyaa, Saba, Bernard Manderick. 2015. Thompson sampling for multi-objective multi-armed bandits problem. *Proceedings*. Presses universitaires de Louvain, 47.

Yahyaa, Saba Q, Madalina M Drugan, Bernard Manderick. 2014. Annealing-pareto multi-objective multi-armed bandit algorithm. *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE, 1–8.

Yang, Zhiguo, Devanathan Sudharshan. 2019. Examining multi-category cross purchases models with increasing dataset scale-An artificial neural network approach. *Expert Systems with Applications* **120** 310–318.

Yianilos, Peter N. 1993. Data structures and algorithms for nearest neighbor search in general metric spaces. *SODA*, vol. 93. 311–21.

Zerfos, Petros, Xiaoqiao Meng, Starsky HY Wong, Vidyut Samanta, Songwu Lu. 2006. A study of the short message service of a nationwide cellular network. *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. 263–268.

Zhang, Shuai, Lina Yao, Aixin Sun, Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* **52**(1) 1–38.

Zhang, Zhiwei, Qifan Wang, Lingyun Ruan, Luo Si. 2014. Preference preserving hashing for efficient recommendation. *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 183–192.

Zhou, Ke, Hongyuan Zha. 2012. Learning binary codes for collaborative filtering. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 498–506.